SUPER 8

Anno 2
n° 8 - Settembre 1985
L. 7.500
Spedizione in abbonamento postale Gruppo III/70

COMMODORE

SUPERCOMMODORE
la rivista per gli utenti dei prodotti
COMMODORE

Una pubblicazione della
J.soft EDITRICE
in collaborazione con
GRUPPO EDITORIALE JACKSON



IL NUOVO COMMODORE 128

FILE RELATIVI: SEMPLICITÀ ED ECONOMIA

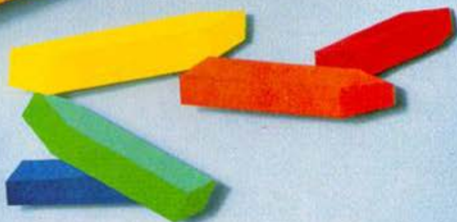
TURBOCOPY PER C64

TURBODISK PER VIC 20 E C64

TANK PER C16 - PLUS/4

C64





dalla biblioteca Jackson informatica per tutti

Rita Bonelli, Luciano Pazzucconi, Fabio Racchi
COMMODORE 16: SEMPRE DI PIÙ
Un libro sul Commodore 16 per approfondire le conoscenze sulla macchina e sul suo BASIC.
cod. 427B Pag. 336
Lire 35.000 Con cassetta

David Lawrence
TECNICHE DI PROGRAMMAZIONE SUL COMMODORE 64
L'arte della buona programmazine alla portata di chiunque possegga un Commodore 64.
cod. 575D Pag. 176
Lire 16.500

Daria Gianni, Carlo Tognoni
MSX: IL BASIC
Il primo libro sul BASIC MSX, che unisce le caratteristiche di un manuale di riferimento a quelle di un buon testo didattico di programmazione.
cod. 417D Pag. 216
Lire 20.500

Brian Lloyd
I TUOI AMICI COMMODORE 16 E PLUS 4
Anche i computer hanno un cuore: impara a programmare con i tuoi amici C16 e Plus 4.
cod. 423B Pag. 168
Lire 16.000

Rodnay Zaks
IL TUO PRIMO COMPUTER
Una semplice introduzione al mondo dei personal orientata ad utenti alla loro prima esperienza con il computer.
cod. 351D Pag. 240
Lire 25.000

ritagliare (o fotocopiare) e spedire in busta chiusa a
GRUPPO EDITORIALE JACKSON – Divisione Libri - Via Rosellini, 12 - 20124 Milano

CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
		Totale	

☐ Pagherò contrassegno al postino il prezzo indicato più **L. 3.000 per contributo fisso spese di spedizione.**

Condizioni di pagamento con esenzione del contributo spese di spedizione:

☐ Allego assegno della Banca ____________

n° ____________

☐ Allego fotocopia del versamento su c/c n. 11666203 a voi intestato

☐ Allego fotocopia di versamento su vaglia postale a voi intestato

Nome ____________
Cognome ____________
Via ____________
Cap ______ Città ______ Prov. ______
Data ______ Firma ______

Spazio riservato alle Aziende. Si richiede l'emissione di fattura
Partita I.V.A. ☐☐☐☐☐☐☐☐☐☐☐

ORDINE MINIMO L. 50.000



GRUPPO EDITORIALE JACKSON

Anno 2 - n° 8 - Settembre 1985

SUPER COMMODORE

J. *soft* s.r.l.
DIREZIONE, REDAZIONE, AMMINISTRAZIONE
Viale Restelli, 5
20124 MILANO
Tel. (02) 68.88.228

DIRETTORE RESPONSABILE:
Pietro Dell'Orco

COORDINAMENTO TECNICO:
Riccardo Paolillo

REDAZIONE:
Mauro Cristuib Grizzi

HANNO COLLABORATO A QUESTO NUMERO
L. Priotto, A. Motta
P. Ravizza, C. Stefani
S. Colombo

GRAFICA E IMPAGINAZIONE:
Margherita La Noce
Ivana Rossi
Raffaella Toffolatti

FOTOCOMPOSIZIONE:
Graphotek - Via Astesani, 16
Tel. 02/64.80.397
20161 MILANO

CONTABILITÀ:
Giulia Pedrazzini
Flavia Bonaiti

AUTORIZZAZIONE ALLA PUBBLICAZIONE:
Tribunale di Milano n° 201
del 14.04.1984

STAMPA:
Litografia del Sole
Albairate (MI)

PUBBLICITÀ
Concessionaria per l'Italia e l'Estero J. Advertising s.r.l.
Viale Restelli, 5
20124 MILANO
Tel. (02) 68.82.895-68.80.606-68.87.233
Tlx. 316213 REINA I

Concessionaria esclusiva per la DIFFUSIONE in Italia e Estero:
SODIP - Via Zuretti, 25
20125 MILANO

Spedizione in abbonamento postale Gruppo III/70

Prezzo della rivista L. 7.500
Numero arretrato L. 15.000
Abbonamento annuo: L. 82.500 (11 numeri con cassetta);
L. 38.500 (11 numeri senza cassetta);
per l'estero: L. 110.000 (11 numeri con cassetta) - L. 50.000 (11 numeri senza cassetta)

I versamenti devono essere indirizzati a:
J. *soft* s.r.l.
Viale Restelli, 5 - 20124 MILANO
mediante emissione di assegno bancario, cartolina vaglia o utilizzando il c/c postale n. 19445204.
Per i cambi di indirizzo indicare, oltre al nuovo, anche l'indirizzo precedente ed allegare alla comunicazione l'importo di L. 500, anche in francobolli

GLI ARTICOLI TRADOTTI SONO TRATTI DALLA RIVISTA COMPUTE! O DA COMPUTE! GAZZETTE - COMPUTE! PUBLICATION, INC.
P.O. BOX 5406,
GREENSBORO, NC 27403 - USA

SCALDA IL JOYSTICK E GASATI CON



SONO IN EDICOLA

RACCOLTA DI GIOCHI E DI UTILITIES PER SPECTRUM

RACCOLTA DI GIOCHI E DI UTILITIES PER C16

RACCOLTA DI GIOCHI E DI UTILITIES PER COMMODORE C64

RACCOLTA DI GIOCHI E DI UTILITIES PER TI99/4A

RACCOLTA DI GIOCHI E DI UTILITIES PER VIC20

RACCOLTA DI GIOCHI E DI UTILITIES PER MSX



GRUPPO EDITORIALE JACKSON

Milano • Londra • Madrid • San Francisco

IL VERO GIOCO COMINCIA ADESSO

EDITORIALE

POINT

Novità

Le novità sono nell'aria.
Dal numero di ottobre, infatti, abbiamo pensato di creare, attorno alle nostre riviste, nuovo e speriamo sempre più ampio interesse.
Vediamo come: prima di tutto la rivista avrà 16 pagine in più; pagine che avranno carattere giornalistico e quindi non listati, ma articoli di vario genere, da leggere per cultura o informazione.
Tra le varie iniziative vi è la realizzazione di due nuove riviste mensili: SuperMSX e SuperApple. Tra voi c'è forse chi occhieggia un sistema più evoluto e uno sguardo alle nuove testate J.*soft* può sempre essere utile.
Ad ottobre parte anche la campagna abbonamenti, tradizionale ed importantissimo evento editoriale per tutte le riviste.
Gli abbonati alle nostre pubblicazioni hanno sofferto un po' in passato, causa un certo disservizio generato da tutta una serie di piccoli problemi, che, sommati, hanno provocato i guai che noi per primi abbiamo lamentato.
Le cose da questo punto di vista si stanno normalizzando e siamo in grado di garantire fin d'ora, per l'immediato futuro e per quanto in nostro potere, un eccellente servizio, poiché la periodicità sarà finalmente rispettata e il servizio abbonati diverrà decisamente più efficiente, grazie a nuove procedure più veloci e più "snelle".
Chiediamo per l'ennesima volta scusa agli abbonati, pregandoli di voler considerare chiusa una spiacevole parentesi.
Il nostro impegno si fa e si farà sempre più concreto, confidando nel costante interesse e apprezzamento che finora ci avete dimostrato ... nonostante tutto!

Pietro Dell'Orco

Per collaborare a SUPERCOMMODORE

La rivista è interessata ad articoli e programmi riguardanti la linea Commodore, di interesse generale, chiari ed esaurienti. Vi proponiamo una piccola "guida", che ha particolarmente lo scopo di rendere più accurata la stesura dei vostri manoscritti: seguendo i nostri consigli si accresceranno le probabilità che le vostre buone idee e i vostri programmi vengano pubblicati.

1 l'angolo superiore sinistro della prima pagina dovrà contenere: nome, cognome, indirizzo, numero telefonico, codice fiscale, data di spedizione, luogo e data di nascita.

2 l'angolo superiore destro della prima pagina dovrà contenere la marca e il tipo di computer al quale il lavoro si riferisce, unitamente ad eventuali espansioni di memoria o periferiche richieste.

3 il titolo dell'articolo, sottolineato, dovrà iniziare a circa due terzi in altezza della prima pagina.

4 le pagine seguenti potranno essere battute normalmente, con la condizione che l'angolo superiore destro contenga un'abbreviazione del titolo e del cognome, unitamente al numero di pagina. Per esempio, Sprite Ed.../Brambilla/2.

5 il testo dell'articolo dovrà essere battuto a macchina con interlinea di uno spazio e mezzo, massimo due spazi; un margine di almeno un centimetro dovrà essere lasciato su entrambi i lati dello scritto.

6 dovranno essere usati fogli in formato UNI A4 (cm. 21 x 29,7) e il testo, scritto in caratteri maiuscoli e minuscoli, dovrà occupare una sola facciata del foglio.

7 nel caso il testo comprenda più fogli, questi dovranno essere uniti con un fermaglio o con un punto metallico aperto.

8 avendo intenzione di spedire più di un articolo, questi dovranno essere inviati separatamente insieme alla rispettiva copia su supporto magnetico.

9 programmi brevi (meno di 20 linee) potranno essere inseriti nel testo, mentre programmi più lunghi dovranno essere listati separatamente. È **ESSENZIALE** per noi disporre di una copia del programma registrata più volte su supporto magnetico, su entrambi i lati dello stesso. È preferibile usare nastri di buona qualità e di lunghezza non eccessiva; la cassetta o il disco dovranno essere etichettati con il nome dell'autore, il titolo dell'articolo, il computer interessato e soprattutto le eventuali espansioni richieste. Come suggerimenti di programmazione si consiglia di usare le istruzioni "CHR$(x)", "TAB(x)", SPC(x)", piuttosto che stringhe di manipolazione del cursore.
Ad esempio, per uno scroll di 5 linee l'istruzione "FORI = 1TO5:PRINT:NEXT". è molto più interpretabile di 5 Q inverse; e, invece di una dozzina di simboli di cursore a destra, perché non usare semplicemente "PRINT SPC (12)"? Un rapido controllo dei programmi per operare queste sostituzioni sarà molto apprezzato da noi e dai lettori.

10 per maggior chiarezza, all'interno dell'articolo è conveniente usare caratteri maiuscoli riferendosi a istruzioni BASIC (esempio RETURN, LIST, RND, PRINT ecc.). Se si desidera evidenziare una parola, è preferibile sottolinearla piuttosto che scriverla in carattere maiuscolo.

11 gli articoli ed i programmi potranno avere qualsiasi lunghezza: da una routine di una sola linea fino a programmi molto complessi.

12 volendo includere diapositive, queste dovranno avere formato 24×36, o 6×6.

13 non prenderemo in considerazione articoli che siano stati sottoposti ad altre case editrici.

14 il materiale non pubblicato non verrà restituito.

15 il compenso per la collaborazione prestata sarà commisurato alla complessità e all'interesse del testo e/o del programma (da un minimo di L. **50.000** a un massimo di L. **300.000**). Il pagamento verrà effettuato in caso di pubblicazione del lavoro.

Inviate idee e programmi a:
SUPERCOMMODORE
Viale Restelli, 5
20124 Milano

e saremo lietissimi di pubblicare i contributi migliori.

La Redazione

La piccola guida del principiante

Che cos'è un programma?

Il solo computer non può compiere alcuna operazione.
Un computer possiede potenzialità, ma, come una macchina senza benzina, senza un programma non può funzionare.
La maggior parte dei programmi pubblicati su **SUPERCOMMODORE** per i computer Commodore sono scritti in un linguaggio per calcolatori chiamato BASIC. Il BASIC è facile da imparare ed è disponibile, di serie, nel VIC 20 e nel Commodore 64.

Programmi BASIC

Ogni mese **SUPERCOMMODORE** pubblica programmi sia per il VIC 20 che per il C64. Tanto per cominciare, se possiedi un VIC 20 copia solo i programmi scritti per la tua macchina, indicati con "versione per VIC 20". Più tardi, quando avrai acquisito esperienza con il BASIC del tuo computer, potrai cercare di copiare e convertire programmi scritti per altri calcolatori. Diversamente dal linguaggio corrente, che può essere variamente interpretato, il BASIC di solito ha un solo modo corretto per indicare qualcosa.
Ogni lettera, carattere o numero ha il suo significato.
Un errore banale è costituito dalla sostituzione del numero "0" con la lettera "O" oppure il carattere minuscolo "l" invece del numero "1" o ancora il carattere maiuscolo "B" con il numero "8".
Devi anche inserire tutta la punteggiatura, i due punti (:) e le virgole, copiando esattamente ciò che appare sulla rivista.
Gli spazi possono essere importanti. Per essere sicuro **copia il listato esattamente** come si presenta.

Le parentesi e i caratteri speciali

L'eccezione per questa regola di copiatura si presenta quando incontrerai indicazioni tra parentesi, quali: "{GIÙ}".
Ogni cosa compresa tra parentesi è un carattere speciale oppure un carattere che non può essere facilmente prodotto con la stampante. Incontrando un carattere di questo tipo fai riferimento alla "Piccola guida per l'input dei programmi".

Le istruzioni DATA

Alcuni programmi contengono una sezione, o delle sezioni, di istruzioni DATA. Queste linee di istruzione forniscono le informazioni di cui il programma ha bisogno.
In alcuni casi le istruzioni DATA costituiscono il programma vero e proprio, in altri contengono codici grafici. Queste linee sono particolarmente soggette agli errori. Se un solo numero in una linea di istruzione DATA è sbagliato, il calcolatore potrebbe "piantarsi" o distruggere il programma. La tastiera e il tasto STOP appaiono inattivi e lo schermo completamente vuoto. Non lasciarti prendere dal panico. Non si è verificato alcun danno.
Per riprendere il controllo devi spegnere il computer e successivamente riaccenderlo. Ciò cancellerà qualsiasi programma presente in memoria, per cui è sempre necessario fare il SAVE del tuo programma prima di comandare il RUN.
Se il computer si ferma, puoi caricare (LOAD) il programma e cercare l'errore.
A volte, quando il programma viene "lanciato", un'istruzione DATA errata può causare un messaggio di errore.
Il messaggio di errore potrebbe riferirsi alla linea di programma che legge (READ) il contenuto delle istruzioni DATA.

Come conoscere il computer

Dovresti prendere confidenza con il computer prima di procedere alla copiatura del programma.
Impara le istruzioni che si usano per memorizzare e richiamare i programmi da nastro o da disco. Dovrai conservare una copia del tuo programma, se non vorrai copiarlo ogni volta che lo devi usare. Impara ad usare le funzioni di "editing" della tua macchina. Come puoi correggere un errore? Puoi sempre ricopiare la linea e in questo caso devi sapere come procedere. Sapresti come inserire i caratteri in "inverse", i caratteri minuscoli e quelli di controllo?
Tutto ciò è spiegato nel manuale del calcolatore.

Un veloce ripasso

1) Copia il programma una linea alla volta, con ordine. Premi RETURN alla fine di ogni linea. Usa il tasto "DEL" per correggere gli errori.
2) Confronta la linea copiata con quella presente nella rivista. Puoi controllare l'intero programma nel caso in cui si presenti un errore quando esegui il RUN.
3) Accertati di aver inserito le istruzioni tra parentesi graffe con gli appropriati caratteri di controllo (fai riferimento alla "Piccola guida per l'input dei programmi" che trovi in questa stessa rivista).

Siamo spiacenti di non poter rispondere singolarmente alle richieste di informazioni circa i programmi, prodotti o servizi apparsi su SUPERCOMMODORE.

OK-LIST

Versione per C16 di **E. Sarcina**

Sappiamo tutti quanto sia difficile digitare correttamente un programma (ovviamente, se non si dispone della cassetta allegata a SUPERCOMMODORE!). Errori di battitura anche banali possono provocare il malfunzionamento del programma stesso, o anche il blocco completo del sistema, obbligandoci a rifare il lavoro.
Ma il punto di forza dei calcolatori è che essi eccellono nei compiti noiosi e che richiedono precisione. Scoprire un errore nella battitura di un listato è appunto un compito noioso, quindi perché non sfruttare il tuo computer per controllare i listati da te digitati?
È proprio questo lo scopo di OK-LIST, il programma che ti permetterà di risparmiare tempo e fatica. Annidato all'interno del tuo computer, OK-LIST controlla automaticamente ogni riga battuta, mostrandoti un numero in cima allo schermo ogni volta che premi il tasto RETURN. Un numero corrispondente nel listato del programma pubblicato ti permette il confronto, ed uno sguardo è sufficiente per avere la conferma di aver digitato la linea correttamente.
OK-LIST è un breve programma in linguaggio macchina residente nel buffer del registratore. Esso rimane in tale zona finché spegni il computer o usi l'unità a cassette.

Come digitare OK-LIST

Vi sono due versioni di OK-LIST, una per C64 e VIC 20 e l'altra per C16 e Plus/4. Digita la versione adatta al tuo computer, facendo attenzione a non commettere errori (per fortuna il listato è molto breve!). Dopo aver finito salvalo su disco o nastro, per lo meno un paio di volte, prima di mandarlo in esecuzione con il comando RUN, perché il programma BASIC, dopo aver memorizzato la routine in linguaggio macchina (le istruzioni DATA), si autocancella.

Come si usa OK-LIST

A questo punto puoi iniziare a battere il tuo primo listato "a prova di errore". Se non hai OK-LIST in memoria, caricalo e attivalo con RUN.
Digita quindi la prima linea di programma: come puoi vedere, in fondo ad essa c'è la parola ":rem" seguita da un numero, che si chiama checksum.
NON COPIARE QUESTA PARTE DELLA LINEA! Ora premi il tasto RETURN, come al solito. In alto a sinistra apparirà il checksum; confrontalo con quello che vedi scritto sulla rivista dopo ":rem". Se i due numeri sono uguali, vuol dire che la linea è stata digitata correttamente, altrimenti controllala, perché evidentemente hai commesso un errore. Facile, no?
OK-LIST non è pignolo per quel che riguarda gli spazi bianchi, dato che non fa caso alla loro presenza. Ciò torna a tuo vantaggio, dal momento che la spaziatura non è, in genere, importante. Ma all'interno delle istruzioni PRINT gli spazi spesso si rivelano necessari, quindi presta attenzione quando digiti questo tipo di istruzione. Un'altra cosa a cui devi badare è che usando le forme abbreviate dei comandi (ad esempio ? per PRINT) il checksum visualizzato non coinciderà con quello presente sul listato. Ciò è tuttavia risolvibile, premendo il tasto RETURN su una linea dopo averla listata: le abbreviazioni saranno ricodificate nella loro forma intera e quindi OK-LIST sarà in grado di svolgere perfettamente la sua funzione. In questo modo, inoltre, avrai la possibilità di controllare interi gruppi di righe in una volta sola.
Ovviamente, nessuno è perfetto, ed anche OK-LIST ha qualche limite: se per caso ti capitasse (ma è un errore poco frequente) di "anagrammare" durante la digitazione una linea (per esempio invece di battere 10 PRINT"VAI", digiti per errore 10 PRINT"VIA"), OK-LIST non se ne accorgerebbe. Inoltre la presenza del carattere SHIFT-SPAZIO modifica il numero di checksum rispetto al carattere SPAZIO normale, anche se la differenza non risulta visibile sul video del tuo computer.
Se intendi sospendere il lavoro di digitazione – o lo hai terminato – puoi salvare su nastro o disco il tuo programma: ricordati comunque di premere RUN/STOP e RESTORE su VIC 20 e C64, oppure RUN/STOP e RESET su C16 e Plus/4 prima di dare il comando SAVE. Al momento di riprendere il lavoro, tuttavia, non potrai più utilizzare OK-LIST, dal momento che qualsiasi operazione su cassetta lo distrugge.
Se vuoi disabilitare OK-LIST in qualunque momento, senza perdere il tuo programma in memoria, premi RUN/STOP e RESTORE su VIC 20 e C64 oppure RUN/STOP e RESET su C16 e Plus/4. Per riattivarlo batti:

SYS 886

Buon lavoro!

OK-LIST
versione per Vic 20 e C64

```
10 PRINT"{CLR}ATTENDERE PREGO..
   ."
11 FORI=886TO1018:READA:CK=CK+A
   :POKEI,A:NEXT
12 IFCK<>14714THENPRINT"{GIU'}E
   RRORE NELLE ISTRUZIONI DATA"
   :END

13 SYS886:PRINT"{CLR}{ 2 GIU'}
   OK-LIST ATTIVATO.":NEW
14 DATA173,034,003,201,150,208
15 DATA001,096,141,151,003,173
16 DATA035,003,141,152,003,169

17 DATA150,141,034,003,169,003
18 DATA141,035,003,169,000,133
19 DATA006,096,032,232,235,133

20 DATA003,134,004,132,005,008
21 DATA201,013,240,017,201,032

22 DATA240,005,024,101,006,133
23 DATA006,165,003,166,004,164
24 DATA005,040,096,169,013,032
25 DATA210,255,165,205,141,251
26 DATA003,206,251,003,169,000
27 DATA133,207,169,019,032,210

28 DATA255,169,018,032,210,255
29 DATA169,058,032,210,255,166
30 DATA006,169,000,133,006,172
31 DATA151,003,192,232,208,006
32 DATA032,095,164,076,235,003
33 DATA032,007,208,169,032,032
34 DATA210,255,032,210.255,173

35 DATA251,003,133,205,076,173
36 DATA003
```

OK-LIST
versione per Plus/4

```
10 PRINT"{CLR}ATTENDERE PREGO..
   ."
11 FORI=886TO1018:READA:CK=CK+A

   :POKEI,A:NEXT
12 IF CK<>17539 THEN PRINT"
   {GIU'}ERRORE NELLE ISTRUZION
   I DATA":END

13 SYS886:PRINT"{CLR}{ 2 GIU'}
   OK-LIST ATTIVATO.":NEW
14 DATA173,036,003,201,150,208
15 DATA001,096,141,151,003,173
16 DATA037,003,141,152,003,169
17 DATA150,141,036,003,169,003

18 DATA141,037,003,169,000,133
19 DATA254,096,032,087,241,133
20 DATA251,134,252,132,253,008
21 DATA201,013,240,017,201,032
22 DATA240,005,024,101,254,133

23 DATA254,165,251,166,252,164
24 DATA253,040,096,169,013,032
25 DATA210,255,165,214,141,251
26 DATA003,206,251,003,169,000
27 DATA133,216,169,019,032,210

28 DATA255,169,018,032,210,255
29 DATA169,058,032,210,255,166
30 DATA254,169,000,133,254,172

31 DATA151,003,192,087,208,006
32 DATA032,205,189,076,235,003

33 DATA032,205,221,169,032,032
34 DATA210,255,032,210,255,173
35 DATA251,003,133,214,076,173
36 DATA003
```

READ &WRITE

La posta dei lettori

Solidarietà e comprensione

Spett. Redazione,

sono un detenuto dell'Asinara, al quale è stato concesso l'uso di un C64 a titolo di studio per l'apprendimento del BASIC. Penso di essere a buon punto, grazie soprattutto alle molte pubblicazioni che trattano il C64 in modo semplice.

Acquisto parecchie riviste di informatica, e solo ultimamente ho iniziato a leggere riviste della J.soft, quali PAPERSOFT e SUPERCOMMODORE, che trovo molto piacevoli ed esaurienti. Ma non è per questo che ho deciso di scrivervi, quanto per segnalare due vostri lettori che, per la loro generosità, mi hanno commosso, convincendomi che una volta pagato il mio debito con la giustizia troverò altre persone come loro che non avranno pregiudizi per il mio passato e spero vorranno offrirmi la stessa amicizia dimostratami dai vostri lettori.

Ebbene, volendo acquistare programmi per il C64, ho risposto a parecchie inserzioni pubblicate dal vostro mensile. Ho avuto possibilità quindi di acquistare a buon prezzo buoni programmi e due vostri lettori, appunto Luciano Ricci di Massa e Marco Romani di Roma, con tanta generosità e gentilezza mi hanno messo a disposizione gratuitamente i loro programmi, che sono veramente tanti, accompagnando il tutto con parole di solidarietà ed amicizia che veramente mi hanno commosso.

Debbo quindi ringraziare anche la vostra rivista che mi ha permesso di acquisire due nuovi amici e, cosa più importante, mi ha fatto comprendere che malgrado tutto non bisogna mai disperare, e bisogna invece avere fiducia nelle persone che ci circondano.

Fausto Marcia

Un bastimento carico di poke

Nel "Protettore di programmi" pubblicato sul numero 3, si accenna ad alcune POKE per disabilitare RUN/STOP + RESTORE e per disabilitare il comando LIST, ma non vengono forniti i valori delle stesse. Potete provvedere?

Piero Marangoni

R. Ecco, per accontentare tutti i lettori che desiderano proteggere i propri programmi, un elenco di locazioni interessanti per il C64, il VIC 20 e il C16-Plus/4:

Funzione		C64	VIC 20	C16-Plus/4
disab.	LIST	775,191	775,223	774,187
riab.	LIST	775,167	775,199	774,110
disab.	SAVE	819,246	818,73	816,136
riab.	SAVE	819,245	818,133	816,164
disab.	LOAD	816,157	816,103	814,239
riab.	LOAD	816,165	816,73	814,74
disab.	RUN/STOP	808,239	808,100	806,103
riab.	RUN/STOP	808,237	808,112	806,101
disab.	RESTORE	792,193	792,7	–
riab.	RESTORE	792,71	792,173	–
disab. la tastiera		649,0	649,0	1343,0
riab. la tastiera		649,10	649,10	1343,10
disab. ripetizione su tutti i tasti		650,64	650,64	1344,64
riab. ripetizione su tutti i tasti		650,128	650,128	1344,128
riab. ripetizione su SPAZIO, DEL, CURS.		650,0	650,0	1344,0
pulisce buffer tastiera		198,0	198,0	239,0
velocità (0-255),	cursore 0=veloce	56325,X	37159,X	–
elimina i numeri di linea nel LIST		22,35	22,35	22,35
riab. numeri di linea		22,25	22,25	22,25

Si noti che se si esegue la POKE di disabilitazione della tastiera in modo diretto, non si sarà più in grado di inserire in seguito la POKE di riabilitazione (essendo appunto disabilitata la tastiera); RUN/STOP+RESTORE funzionerà ancora comunque, anche se è stato disabilitato, e ripristinerà il funzionamento della tastiera. Inoltre, accadono cose strane, se si esegue la POKE22,35 per disabilitare la stampa dei numeri di linea. Ad esempio, il computer non stamperà più stringhe racchiuse tra virgolette (es: PRINT"CIAO", in modo diretto), mentre stamperà ancora le variabili di stringa. Altre POKE creano effetti collaterali: ad esempio, se si modifica la velocità del cursore, il jiffy clock – l'orologio interno al computer – si mette anch'esso a correre o a rallentare.

Ecco infine una interessante serie di POKE: se accidentalmente digitate NEW e volete recuperare il programma in memoria, inserite in modo diretto quanto segue.

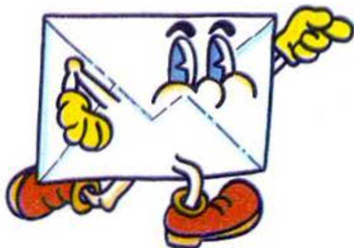
Per il C64:

POKE2050,1 : SYS42291 : POKE45, PEEK(34) : POKE46, PEEK(35) : CLR

Per il VIC 20:

POKE4098,1 : SYS50483 : POKE45, PEEK(34) : POKE46, PEEK(35) : CLR

Se avete collegata al VIC 20 un'espansione di memoria da 8 Kbyte o più, dovete modificare la prima POKE in POKE4610,1.







La nuova generazione di personal computer

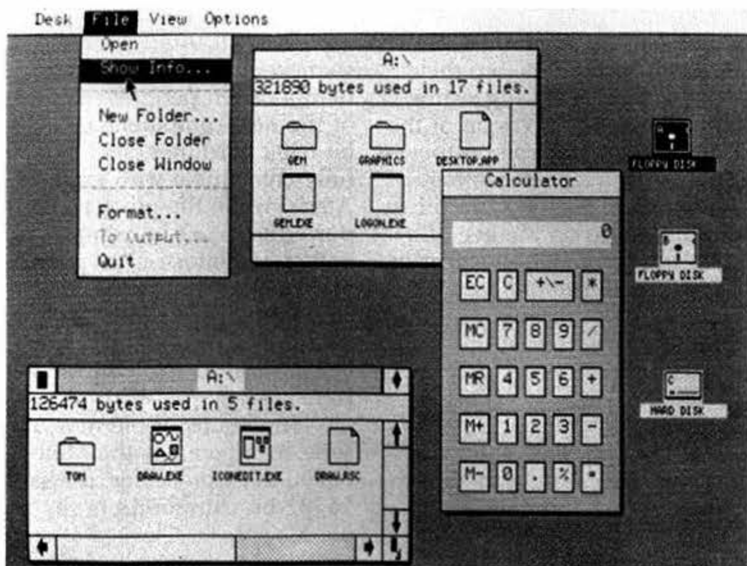
di **T. R. Halfhill**
trad. ed adatt.
di **M. Cristuib Grizzi**

Fino ad ora IBM ed Apple hanno dominato incontrastate il settore "alto" del mercato dei personal computer, lasciando il settore "basso" a Commodore, Atari e Sinclair. Dopo il "Consumer Electronic Show", tenutosi a Las Vegas in Gennaio, queste diverse fasce di mercato hanno assunto la tendenza a sovrapporsi e poi scomparire, a causa dell'introduzione di nuove macchine dotate di prestazioni fino a ieri tipiche della fascia "alta", a prezzi appartenenti invece alla fascia "bassa" del mercato.



I computer della Atari, colosso americano in diretta concorrenza con la Commodore, sono in Italia ancora molto poco diffusi, mentre negli Stati Uniti occupano un settore notevole del mercato. Viste le novità – e soprattutto i prezzi – presentate possiamo aspettarci una larga diffusione dei prodotti Atari anche nel nostro Paese.
In tutto, Atari ha presentato sei nuovi computer e più di una dozzina di periferiche: quattro dei nuovi computer sono macchine a otto bit, basate sul 6502, compatibili con i precedenti computer Atari. Le altre due novità – delle quali appunto parleremo – sono potenti computer a 16/32 bit con un sistema operativo molto simile a quello dell'Apple Macintosh, e costituiscono la serie ST.
Il cervello di questi computer è il microprocessore Motorola 68000, lo stesso presente nel Macintosh. Con una frequenza di clock di otto megahertz il 68000 ha una velocità di esecuzione circa doppia del chip 8088 a 8/16 bit che equipaggia gli IBM PC, PC-XT e PCjr.
Gli Atari della nuova serie ST, il 130ST e il 520ST, sono mutualmente compatibili ed offrono molte caratteristiche comuni, tra le quali 192 Kbyte di ROM, espandibili a 320 Kbyte tramite una cartridge, 512 colori diversi, risoluzione di 320x200 pixel a 16 colori, 640x200 pixel a quattro colori e 640x400 pixel con un solo colore. Essi sono inoltre dotati di interfaccia parallela Centronics, seriale RS-232 standard, interfaccia per drive a floppy disk ed hard disk; è presente inoltre un'interfaccia MIDI (Musical Instruments Digital Interface) per il collegamento a tastiere e sintetizzatori di suono esterni. Due porte joystick (una delle quali è usata per il collegamento del mouse), uscita TV ed RGB, chip sonoro a tre voci con forme d'onda

ed inviluppo variabili, tastiera a 94 tasti con tastierino numerico separato e dieci speciali tasti funzione.
Il loro sistema operativo è stato denominato TOS (Tramiel Operating System), mentre le routine ROM che presiedono alla parte grafica sono state denominate GEM (Graphic Environment Manager): il GEM è un'interfaccia di sistema operativo "tipo Macintosh", sviluppata inizialmente per computer sotto MS-DOS, ed il cui brevetto è stato ceduto alla Atari dalla Digital Research. Il GEM consente, proprio come nel Macintosh, di lavorare con icone, menu a scorrimento, finestre e un puntatore controllato dal mouse. Comprende inoltre un orologio in tempo reale, grafica vettorizzata in alta risoluzione ed animazioni sul tipo degli Sprite.
Le uniche differenze tra l'Atari 130ST e il 520ST stanno nell'ammontare di RAM disponibile e nel prezzo. Il 130ST dispone di 128 Kbyte di RAM e verrà a costare – per quanto si sappia al momento della stesura di questo articolo, in Aprile – intorno alle ottocentomila lire, mentre il 520ST, con 512 Kbyte di RAM, dovrebbe costare intorno al milione e centomila lire.
Se pensate forse che questi prezzi non siano credibili, viste le prestazioni offerte, sentite ora ciò che riguarda le periferiche: per la serie ST è stato annunciato un drive per microfloppy a 3 pollici e mezzo, che verrà a costare intorno alle – udite, udite – duecentomila lire. Questo drive usa gli stessi dischi in standard Sony usati dal Macintosh: i dischi a faccia singola possono contenere fino a 250 Kbyte, mentre la versione a doppia faccia (dischetti dal costo leggermente più elevato) fino a mezzo megabyte. Ancora più incredibile è l'annunciato hard disk per la stessa serie ST (e, con il marchio Atarisoft, anche per altri computer), che dovrebbe consistere in un disco rigido da 15 megabyte con il costo di circa ottocentomila lire.
L'interfaccia per hard disk implementata sulla serie ST è in grado di trasferire dati a 1.33 megabyte al secondo, circa 100 volte più velocemente di un tipico hard disk per Macintosh. Ciò significa che è possibile riempire tutta la RAM di 512 Kbyte del 520ST, con un programma o con dei dati, in meno di mezzo secondo...
Non solo i computer della serie ST sono molto più potenti degli home computer attualmente esistenti, ma addirittura superano potenzialmente le capacità della maggior parte dei computer professionali a 16 bit oggi sul mercato. Promettendo l'equivalente di un "grosso Mac" a colori, e a un terzo del suo prezzo, l'Atari si prepara a ridefinire l'intero mercato degli home e dei personal.
Con questa "valanga" di grosse novità Atari in arrivo, la Commodore si prepara a contrattaccare con alcuni nuovi prodotti.
Ancora scottata dall'insuccesso del Plus/4, la Commodore ha finalmente



Ecco una tipica schermata GEM in versione MS-DOS: la versione ST è virtualmente identica. Si vedono il menu a scorrimento, le icone e le finestre a sovrapposizione.

pensato bene di rendere compatibili con il C64 i propri nuovi "desktop computer".
Il nuovo Commodore 128 contiene infatti quasi tre diversi computer in un unico chassis: contiene un normale C64, con microprocessore 6510, chip video VIC-II e chip sonoro SID, e 64 Kbyte di RAM; contiene poi un computer con microprocessore 8502 (compatibile 6502/6510), 128 Kbyte di RAM e video a 80/40 colonne, ed infine contiene un computer con microprocessore Z80A, 128 Kbyte di RAM e schermo a 80 colonne, compatibile con il software scritto per il sistema operativo CP/M.
Il C128 è espandibile fino a 512 Kbyte, per incrementi successivi di 128 Kbyte, anche se la memoria aggiuntiva può essere usata solo come un disco RAM e non come memoria continua per programmi. Da alcune indicazioni per ora disponibili sembra che il C128 potrà costare intorno alle sette-ottocentomila lire.
Ovviamente, il C128 è molto di più che una semplice versione a 128 Kbyte del C64 (che era stata per altro precedentemente anticipata dalla Commodore). Appena acceso, il computer si regola automaticamente in modo 128 Kbyte con microprocessore 8502: è possibile selezionare il video tra quaranta e ottanta colonne, ma grafica e Sprite sono disponibili solo nel modo a quaranta colonne.
Per passare sotto CP/M occorre caricare il sistema operativo da un disco che viene fornito insieme al computer, mentre per passare in modo C64 basta digitare "GO 64". Lo schermo scompare per qualche secondo, e viene quindi visualizzato il classico messaggio generato dal C64 appena acceso.
Una volta entrati in modo C64 non c'è maniera di passare agli altri modi, se non spegnendo e riaccendendo la macchina. È questo, se si vuole, lo scotto da pagare per avere la compatibilità totale con il C64: l'implementazione di un comando tipo "GO 128" avrebbe infatti richiesto una modifica al sistema operativo. Per la stessa ragione il video a ottanta colonne non è disponibile nel modo C64.
Il C128 dispone di due diversi linguaggi di programmazione: il solito BASIC 2.0 del C64 ed il BASIC 7.0 nel modo a 128 Kbyte. Quest'ultimo linguaggio è il più potente BASIC Commodore fino ad ora realizzato, con gli stessi comandi per la gestione del drive del BASIC 4.0 implementato sul CBM 8032, unitamente a comandi per grafica e suono sul tipo di quelli della cartuccia "Super Expander 64".
In modo CP/M è disponibile una gran quantità di linguaggi orientati all'uso del drive, compresi vari compilatori ed assemblatori.
Altre caratteristiche del C128 sono la tastiera a 92 tasti con tastierino numerico separato, due set di tasti cursore, quattro tasti funzione speciali e nuovi tasti etichettati ALT, ESC, TAB, CAPS LOCK, HELP, LINE FEED, 40/80 DISPLAY, NO SCROLL.
Sono presenti uscite sia TV che RGB con audio, sedici colori, monitor di linguaggio macchina e le stesse porte di comunicazione con l'esterno già implementate sul C64.
Il C128 non è certamente tanto potente ed interessante quanto i nuovi Atari ST, ma nasce dalla stessa filosofia: offre un computer "tutto in uno", che permette

una gran varietà di applicazioni casalinghe, personali e commerciali, a un prezzo ragionevole.
Con la possibilità di diventare un C64, il C128 viene a disporre di una grossa quantità di software didattico, di giochi e di applicazioni di piccola gestione commerciale. In modo 128 Kbyte con il BASIC 7.0 diventa molto più facile da programmare di un C64. In modo CP/M sono infine disponibili una gran quantità di programmi applicativi e "seri". Benché il CP/M sia ormai stato superato in diffusione dal MS-DOS dell'IBM PC, con tutta la schiera dei PC compatibili, rimane pur sempre adeguato per impieghi di piccola gestione aziendale.
Insieme al C128 nasce una serie di nuove periferiche, tra cui il nuovo disk drive 1571. Trovandosi in modo C64, il 1571 si comporta esattamente come il "solito" 1541, immagazzinando 140 Kbyte su ogni disco da 5 pollici e un quarto, con una velocità di trasferimento dati molto bassa. Passando tuttavia in modo 128 Kbyte, il 1571 diventa molto più veloce ed opera anche su doppia faccia, permettendo una capacità di 350 Kbyte per disco. In modo CP/M, infine, il 1571 diventa ancora più veloce (circa 12 volte più veloce del 1541) ed aumenta la capacità di immagazzinamento dati a 410 Kbyte per disco. Inoltre, in modo CP/M il 1571 è in grado di leggere dischi in formato CP/M IBM 34, compresi dischi Osborne e Kaypro. Come se ciò non bastasse, il 1571 è collegabile direttamente al C64, C16 e Plus/4, anche se in questo modo funziona unicamente come un normale 1541.
Per il C128 sono stati annunciati anche due nuovi monitor: il 1901 monocromatico, dedicato ad applicazioni commerciali a 80 colonne, e il 1902 a colori, che riesce a visualizzare le ottanta colonne in un formato leggibile a colori. Al momento non sono ancora stati annunciati i prezzi di queste periferiche.
Un'altra novità Commodore è il portatile LCD, con un display a cristalli liquidi di 80 colonne per 16 righe, dalla velocità di visualizzazione elevatissima. In posizione chiusa il video costituisce un coperchio di protezione della tastiera a 72 tasti. Le caratteristiche di questo computer comprendono un microprocessore a otto bit 65C102, 32 Kbyte di RAM e 96 Kbyte di ROM, porta seriale standard RS-232, porta parallela standard Centronics, interfaccia per lettore di codice a barre, porta seriale compatibile con le periferiche standard del C64 e il drive 1571, porta di espansione per la RAM, modem a 300 baud con possibilità di chiamata e risposta automatica, BASIC 3.6, monitor di linguaggio macchina, otto programmi residenti su ROM. Questi ultimi consistono in word processor, blocco per appunti, foglio elettronico, gestione file, emulatore di terminale, calcolatrice, archivio, e rubrica indirizzi.
Abbastanza piccolo per poter essere contenuto in una borsa, l'LCD pesa poco meno di due chili e trae alimentazione da quattro batterie oppure da una sorgente esterna, mantenendo il contenuto della RAM anche quando il computer è spento.
Il suo prezzo stimato è di circa un milione e duecentomila lire, molto competitivo rispetto agli altri portatili già sul mercato.
La "grossa" novità Commodore, comunque, che pare venga tenuta nel cassetto per essere presentata verso la fine dell'anno, è il cosiddetto – o meglio la cosiddetta – "Amiga Lorraine".
Un prototipo di questo avanzato computer era stato mostrato al Consumer Electronic Show dello scorso anno, quando l'Amiga era ancora una società indipendente in cerca di finanziatori.
Avendola ora assorbita, la Commodore ha acquistato quindi i diritti di produzione del Lorraine. Esso è molto simile, come potenza, alla serie ST dell'Atari – usa lo stesso microprocessore –, ma rispetto a questa potrà offrire molte caratteristiche aggiuntive, quali sintetizzatore di voce, grafica e suono ancora più perfezionati ed una grande espandibilità.
La Commodore ha confermato che questo "supermicro" è in via di completamento, e dovrebbe essere commercializzato a fine anno ad un prezzo intorno ai mille dollari (circa due milioni di lire).
Apple ed IBM stanno per il momento a guardare, e non senza qualche preoccupazione.



L'Atari 520ST: microprocessore Motorola 68000, 512 colori, sistema operativo "tipo Macintosh", 512 Kbyte di RAM, ad un prezzo incredibile. Il 130ST è funzionalmente ed esteticamente identico, ma dispone di "solo" 128 Kbyte di RAM.



Il Commodore LCD è un portatile con schermo a cristalli liquidi 80x16, 32 Kbyte di RAM, modem interno e otto programmi su ROM.

MLX per VIC 20 e C64 (versione 2.0 per C64)

di **C. Brannon**
trad. e adatt.
di **M. Cristuib Grizzi** e **F. Stella**

MLX è un programma che permette di inserire listati in linguaggio macchina esenti in modo assoluto da errori e senza la perdita di tempo del dover battere e controllare lunghe sequenze di istruzioni DATA.

Molti dei nostri listati di programmi in linguaggio macchina hanno il formato MLX (li riconoscete dal fatto che sono esclusivamente numerici) e richiedono quindi che MLX sia caricato in memoria ed eseguito prima della battitura del listato.

MLX vi chiederà l'indirizzo della locazione di partenza e quello della locazione finale del programma da caricare. Questi valori sono sempre indicati nell'articolo che accompagna il listato. MLX vi segnala automaticamente gli errori di battitura MENTRE STATE DIGITANDO IL LISTATO e vi chiede di reinserire la linea errata. L'ultimo numero battuto di ogni linea ràppresenta il checksum e viene visualizzato in reverse.

Sono inoltre disponibili altri comandi, quali:

SHIFT-N nuovo indirizzo: permette di cambiare l'indirizzo della linea che volete battere ed è utile nel caso si inseriscano i listati in più parti.

SHIFT-D display: lista i dati inseriti tra due indirizzi di inizio e fine.

SHIFT-L load: carica un file da nastro o

SHIFT-S save: salva su nastro o disco un file in formato MLX.

Una volta battuto il listato, e salvatolo tramite MLX, si potrà caricare direttamente il programma con un'istruzione LOAD "nome del file",1,1 per il registratore, oppure LOAD "nome del file",8,1 per l'unità a dischi, salvo diverse indicazioni date negli articoli corrispondenti ai vari programmi. Il programma sarà generalmente mandato in esecuzione con una SYS (indirizzo di partenza).

La versione originaria di MLX per C64 ha subito, dalla prima pubblicazione, diversi ritocchi e miglioramenti, fino alla versione 2.0 qui presentata.

È stato in primo luogo modificato il colore di fondo dello schermo e del bordo, in modo da risultare meno stancante alla vista; sono state ulteriormente perfezionate le routine in linguaggio macchina di salvataggio e caricamento dei programmi; è cambiata la forma del cursore e, cosa più importante, è stato aggiunto un tastierino numerico per gli utenti che si trovano più a loro agio con una diversa disposizione dei tasti.

Oltre a poter usare i soliti tasti numerici, sono stati ridefiniti alcuni tasti alfabetici, in modo da generare ugualmente dei numeri. I tasti ridefiniti sono i seguenti:

U I O		7 8 9
H J K L	diventano	0 4 5 6
M , .		1 2 3

Le persone abituate ad usare tali tastierini accresceranno senz'altro, in brevissimo tempo, la velocità di battitura dei programmi in linguaggio macchina.

MLX versione per VIC 20

```
100 PRINT"{CLR}{PUR}";CHR$(142);CHR$(8);
                                :rem 181
101 POKE788,194:REM DISABILITA RUN/STOP
                                :rem 144
110 PRINT"{RVS}{ 14 SPAZI}"     :rem 117
120 PRINT"{RVS} {DES}{OFF}[<*>]£{RVS}
    {DES} {DES}{ 2 SPAZI}[<*>]{OFF}[<*>]
    £{RVS}£{RVS} "              :rem 191
130 PRINT"{RVS} {DES} [<G>]{DES}
    { 2 DES} {OFF}£{RVS}£[<*>]{OFF}[<*>]
    {RVS} {OFF}"                :rem 122
140 PRINT"{RVS}{ 14 SPAZI}"     :rem 120
200 PRINT"{ 2 GIU'}{PUR}{BLK}UN PROGRAMMA
     PER L'IN-TRODUZIONE{ 2 SPAZI}DI ROUT
    INEIN LINGUAGGIO MACCHINA"; :rem 202
205 PRINT"A PROVA DI ERRORE{ 5 GIU'}"
                                :rem 226
210 PRINT"{BLK}{ 4 SU}INDIRIZZO DI PARTEN
    ZA":INPUTS:F=1-F:C$=CHR$(31+119*F)
                                :rem 52
220 IFS<256ORS>32767THENGOSUB3000:GOTO210
                                :rem 2
225 PRINT:PRINT:PRINT:PRINT     :rem 123
230 PRINT"{BLK}{ 3 SU}INDIRIZZO CONCLUSIV
    O":INPUTE:F=1-F:C$=CHR$(31+119*F)
                                :rem 170
240 IFE<256ORE>32767THENGOSUB3000:GOTO230
                                :rem 234
250 IFE<STHENPRINTC$;"{RVS}INDIRIZZO CONC
    LUSIVO <INDIRIZZO DI PARTENZA
    { 3 GIU'}":GOSUB1000:GOTO230:rem 27
260 PRINT:PRINT:PRINT           :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S  :rem 56
310 PRINTRIGHT$("0000"+MID$(STR$(AD),2),5
    );":";:FORJ=1TO6            :rem 234
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
                                :rem 228
390 IFN=-211THEN710             :rem 62
400 IFN=-204THEN790             :rem 64
410 IFN=-206THENPRINT:PRINT"{GIU'}INSERIS
    CI UN NUOVO{ 4 SPAZI}INDIRIZZO";:INPU
    TZZ                         :rem 42
415 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
    {RVS}ESCE DAL CAMPO DI VA-" :rem 150
416 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
    {RVS}LORI INDICATO":GOSUB1000:GOTO410
                                :rem 105
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
                                :rem 238
420 IFN<>-196THEN480            :rem 133
430 PRINT:INPUT"LISTATO:DA";F:PRINT,"
    { 2 SIN}A";:INPUTT          :rem 29
440 IFF<SORF>EORT<SORT>ETHENPRINT"MINIMO"
    ;S;"{SIN}, MASSIMO";E:GOTO430
                                :rem 36
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("000
    0"+MID$(STR$(I),2),5);":";  :rem 30
455 FORK=0TO5:N=PEEK(I+K):IFK=3THENPRINTS
    PC(10);                     :rem 34
457 PRINTRIGHT$("00"+MID$(STR$(N),2),3);"
    ,";                         :rem 157
460 GETA$:IFA$>""THENPRINT:PRINT:GOTO310
                                :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
    T:GOTO310                   :rem 50
480 IFN<0THENPRINT:GOTO310      :rem 168
490 A(J)=N:NEXTJ                :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
    SUM=(CKSUM+A(I))AND255:NEXT :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(20)
                                :rem 234
515 IFN=CKSUMTHEN530            :rem 255
520 PRINT:PRINT"LA RIGA E' STATA INSE-RIT
    A IN MANIERA ERRATA"        :rem 39
525 PRINT"RI-INSERIRLA":PRINT:GOSUB1000:G
    OTO310                      :rem 167
530 GOSUB2000                   :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT
                                :rem 80
550 AD=AD+6:IFAD<ETHEN310       :rem 212
560 GOTO710                     :rem 108
570 N=0:Z=0                     :rem 88
```

```
580 PRINT"[<+>]";                     :rem  79
581 GETA$:IFA$=""THEN581              :rem  95
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44
    ORA=32THEN670                     :rem 229
590 IFA>128THENN=-A:RETURN            :rem 137
600 IFA<>20THEN630                    :rem  10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
    {SIN} {SIN}";:GOTO690             :rem 172
620 GOTO570                           :rem 109
630 IFA<48ORA>57THEN580               :rem 105
640 PRINTA$;:N=N*10+A-48              :rem 106
650 IFN>255THENA=20:GOSUB1000:GOTO600
                                      :rem 229
660 Z=Z+1:IFZ<3THEN580                :rem  71
670 IFZ=0THENGOSUB1000:GOTO570        :rem 114
680 PRINT",";:RETURN                  :rem 240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
                                      :rem 149
692 FORI=1TO3:T=PEEK(S%-I)            :rem  68
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
                                      :rem 205
700 PRINTLEFT$("{ 3 SIN}",I-1);:RETURN
                                      :rem   7
710 PRINT"{CLR}{RVS}*** SAVE ***
    { 3 GIU'}"                        :rem 236
720 INPUT"{GIU'} NOME DEL FILE";F$
                                      :rem 199
730 PRINT:PRINT"{ 2 GIU'}{RVS}N{OFF}ASTRO
    O {RVS}D{OFF}ISCO: (N/D)"         :rem 128
740 GETA$:IFA$<>"N"ANDA$<>"D"THEN740
                                      :rem  30
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                      :rem 158
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                  :rem   3
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                      :rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                                :rem  69
765 POKE254,S/256:POKE253,S-PEEK(254)*256
    :POKE780,253                      :rem  12
766 POKE782,E/256:POKE781,E-PEEK(782)*256
    :SYS65496                         :rem 124
770 IF(PEEK(783)AND1)OR(ST AND191)THEN780
                                      :rem 111
775 PRINT"{GIU'}OPERAZIONE CONCLUSA.":END
                                      :rem 152
780 PRINT"{GIU'}ERRORE DI REGISTRAZIO-NE.
    RIPROVA.":IFDV=1THEN720           :rem   8
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO720              :rem 103
782 GOTO720                           :rem 115
790 PRINT"{CLR}{RVS}*** LOAD ***
    { 2 GIU'}"                        :rem 212
800 INPUT"{ 2 GIU'} NOME DEL FILE";F$
                                      :rem 215
810 PRINT:PRINT"{ 2 GIU'}{RVS}N{OFF}ASTRO
    O {RVS}D{OFF}ISCO: (N/D)"         :rem 127
820 GETA$:IFA$<>"N"ANDA$<>"D"THEN820
                                      :rem  28
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                      :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                  :rem   2
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                      :rem 107
845 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                                :rem  70
850 POKE780,0:SYS65493                :rem  11
860 IF(PEEK(783)AND1)OR(ST AND191)THEN870
                                      :rem 111
865 PRINT"{GIU'}OPERAZIONE CONCLUSA.":GOT
    O310                              :rem 142
870 PRINT"{GIU'}ERRORE DI CARICAMENTO.RIP
    ROVA.{GIU'}":IFDV=1THEN800        :rem  43
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO800              :rem 102
1000 REM CICALINO                     :rem 231
1001 POKE36878,15:POKE36874,190       :rem 206
1002 FORW=1TO300:NEXTW                :rem 117
1003 POKE36878,0:POKE36874,0:RETURN
                                      :rem  74
2000 REM CAMPANELLO                   :rem 130
2001 FORW=15TO0STEP-1:POKE36878,W:POKE368
     76,240:NEXTW                     :rem  22
2002 POKE36876,0:RETURN               :rem 119
3000 PRINTC$;"{RVS}NON IN PAGINA ZERO O
     { 2 SPAZI}SU ROM{ 4 GIU'}":GOTO1000
                                      :rem  23
```

MLX versione 2.0 per C64

```
100 PRINT"{CLR}{CYN}";CHR$(142);CHR$(8);:
    POKE53280,0:POKE53281,0           :rem  71
101 POKE788,52:REM DISABILITA RUN/STOP
                                      :rem  89
110 PRINT"{RVS}{ 40 SPAZI}";          :rem 176
120 PRINT"{RVS}{ 15 SPAZI}{DES}{OFF}[<*>]
    ~{RVS}{DES} {DES}{ 2 SPAZI}[<*>]{OFF}
    [<*>]~{RVS}~{RVS}{ 13 SPAZI}";
                                      :rem 250
130 PRINT"{RVS}{ 15 SPAZI}{DES}[<N>][<H>]
    {DES} { 2 DES} {OFF}~{RVS}~[<*>]{OFF}
    [<*>]{RVS}{ 13 SPAZI}";           :rem 220
140 PRINT"{RVS}{ 40 SPAZI}"           :rem 120
200 PRINT"{ 2 GIU'}{PUR}EDITOR DI CODICE
    MACCHINA VERSIONE 2.0{ 3 GIU'}"
                                      :rem 126
210 PRINT"[<5>]{ 2 SU} INDIRIZZO DI PARTE
    NZA{ 2 SPAZI}";:INPUTS:F=1-F:C$=CHR$(
    31+119*F)                         :rem  85
220 IFS<256OR(S>40960ANDS<49152)ORS>53247
    THENGOSUB3000:GOTO210             :rem 235
225 PRINT:PRINT:PRINT                 :rem 180
230 PRINT"[<5>]{ 2 SU} INDIRIZZO CONCLUSI
    VO{ 3 SPAZI}";:INPUTE:F=1-F:C$=CHR$(3
    1+119*F)                          :rem  92
240 IFE<256OR(E>40960ANDE<49152)ORE>53247
    THENGOSUB3000:GOTO230             :rem 183
250 IFE<STHENPRINTC$;"{RVS}INDIRIZZO CONC
    LUSIVO<INDIRIZZO INIZIALE"        :rem 200
255 IFE<STHENGOSUB1000:GOTO230        :rem 119
260 PRINT:PRINT:PRINT                 :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S:POKEV+21,0
                                      :rem 225
310 A=1:PRINTRIGHT$("0000"+MID$(STR$(AD),
    2),5);":";:FORJ=ATO6              :rem 227
315 FORJ=ATO6                         :rem  33
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
                                      :rem 228
390 IFN=-211THEN710                   :rem  62
400 IFN=-204THEN790                   :rem  64
410 IFN=-206THENPRINT:INPUT"{GIU'} INSERI
    RE IL NUOVO INDIRIZZO";ZZ         :rem  13
414 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
    {RVS}ESCE DAL CAMPO DI VALORI INDICAT
    O"                                :rem 105
```

```
415 IFN=-206THENIFZZ<SORZZ>ETHENGOSUB1000
    :GOTO410                        :rem 202
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
                                    :rem 238
420 IFN<>-196THEN480                :rem 133
430 PRINT:INPUT"LISTATO:DA";F:PRINT"
    { 9 SPAZI}A";:INPUTT            :rem 183
440 IFF<SORF>EORT<SORT>ETHENPRINT"MINIMO"
    ;S;" MASSIMO";E;"![<5>]":GOTO430
                                    :rem 147
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("000
    0"+MID$(STR$(I),2),5);":";      :rem 30
451 FORK=0TO5:N=PEEK(I+K):PRINTRIGHT$("00
    "+MID$(STR$(N),2),3);",";       :rem 66
460 GETA$:IFA$>""THENPRINT:PRINT:GOTO310
                                    :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
    T:GOTO310                       :rem 50
480 IFN<0THENPRINT:GOTO310          :rem 168
490 A(J)=N:NEXTJ                    :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
    SUM=(CKSUM+A(I))AND255:NEXT     :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(146
    );                              :rem 94
511 IFN=-1THENA=6:GOTO315           :rem 254
515 PRINTCHR$(20):IFN=CKSUMTHEN530
                                    :rem 122
520 PRINT:PRINT"{RED}LA LINEA E' STATA IN
    SERITA IN MANIERA"              :rem 157
525 PRINT"ERRATA. RIPETERE[<5>]":PRINT:GO
    SUB1000:GOTO310                 :rem 27
530 GOSUB2000                       :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:POKE54
    272,0:POKE54273,0               :rem 227
550 AD=AD+6:IFAD<ETHEN310           :rem 212
560 GOTO710                         :rem 108
570 N=0:Z=0                         :rem 88
580 PRINT"[<~>]";                   :rem 81
581 GETA$:IFA$=""THEN581            :rem 95
582 AV=-(A$="M")-2*(A$=",")-3*(A$=".")-4*
    (A$="J")-5*(A$="K")-6*(A$="L")
                                    :rem 41
583 AV=AV-7*(A$="U")-8*(A$="I")-9*(A$="O"
    ):IFA$="H"THENA$="0"            :rem 134
584 IFAV>0THENA$=CHR$(48+AV)        :rem 134
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44
    ORA=32THEN670                   :rem 229
590 IFA>128THENN=-A:RETURN          :rem 137
600 IFA<>20THEN630                  :rem 10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
    {SIN} {SIN}";:GOTO690           :rem 172
620 GOTO570                         :rem 109
630 IFA<48ORA>57THEN580             :rem 105
640 PRINTA$;:N=N*10+A-48            :rem 106
650 IFN>255THENA=20:GOSUB1000:GOTO600
                                    :rem 229
660 Z=Z+1:IFZ<3THEN580              :rem 71
670 IFZ=0THENGOSUB1000:GOTO570      :rem 114
680 PRINT",";:RETURN                :rem 240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
                                    :rem 149
691 FORI=1TO3:T=PEEK(S%-I)          :rem 67
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
                                    :rem 205
700 PRINTLEFT$("{ 3 SIN}",I-1);:RETURN
                                    :rem 7
710 PRINT"{CLR}{RVS}*** SAVE ***
    { 3 GIU'}"                      :rem 236
715 PRINT"{ 2 GIU'}PREMERE {RVS}RETURN
    {OFF} PER USCIRE DAL SAVE{GIU'}"
                                    :rem 103
720 F$="":INPUT"{GIU'}NOME DEL FILE";F$:I
    FF$=""THENPRINT:PRINT:GOTO310
                                    :rem 42
730 PRINT:PRINT"{ 2 GIU'}{RVS}N{OFF}ASTRO
    O {RVS}D{OFF}ISCO: (N/D)"       :rem 128
740 GETA$:IFA$<>"N"ANDA$<>"D"THEN740
                                    :rem 30
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$:
    OPEN15,8,15,"S"+F$:CLOSE15      :rem 212
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                :rem 3
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                    :rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                              :rem 69
765 K=S:POKE254,K/256:POKE253,K-PEEK(254)
    *256:POKE780,253                :rem 17
766 K=E+1:POKE782,K/256:POKE781,K-PEEK(78
    2)*256:SYS65496                 :rem 235
770 IF(PEEK(783)AND1)OR(191ANDST)THEN780
                                    :rem 111
775 PRINT"{GIU'}FATTO.{GIU'}":GOTO310
                                    :rem 201
780 PRINT"{GIU'}ERRORE NEL SAVE-RIPROVA!"
    :IFDV=1THEN720                  :rem 104
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO720            :rem 103
790 PRINT"{CLR}{RVS}*** LOAD ***
    { 2 GIU'}"                      :rem 212
795 PRINT"{ 2 GIU'}PREMERE {RVS}RETURN
    {OFF} PER USCIRE DAL LOAD{GIU'}"
                                    :rem 96
800 F$="":INPUT"{ 2 GIU'}NOME DEL FILE";F
    $:IFF$=""THENPRINT:GOTO310      :rem 115
810 PRINT:PRINT"{ 2 GIU'}{RVS}N{OFF}ASTRO
    O {RVS}D{OFF}ISCO: (N/D)"       :rem 127
820 GETA$:IFA$<>"N"ANDA$<>"D"THEN820
                                    :rem 28
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                    :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
    ):POKE782,ZK/256                :rem 2
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
    T$):SYS65469                    :rem 107
845 POKE780,1:POKE781,DV:POKE782,1:SYS654
    66                              :rem 70
850 POKE780,0:SYS65493              :rem 11
860 IF(PEEK(783)AND1)OR(191ANDST)THEN870
                                    :rem 111
865 PRINT"{GIU'}FATTO.":GOTO310     :rem 184
870 PRINT"{GIU'}ERRORE NEL LOAD-RIPETI!
    {GIU'}":IFDV=1THEN800           :rem 19
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
    ;E2$:CLOSE15:GOTO800            :rem 102
1000 REM CICALINO                   :rem 231
1001 POKE54296,15:POKE54277,45:POKE54278,
     165                            :rem 207
1002 POKE54276,33:POKE54273,6:POKE54272,5
                                    :rem 42
1003 FORT=1TO200:NEXT:POKE54276,32:POKE54
     273,0:POKE54272,0:RETURN       :rem 202
2000 REM CAMPANELLO                 :rem 130
2001 POKE54296,15:POKE54277,0:POKE54278,2
     47                             :rem 152
2002 POKE54276,17:POKE54273,40:POKE54272,
     0                              :rem 86
2003 FORT=1TO100:NEXT:POKE54276,16:RETURN
                                    :rem 57
3000 PRINTC$;"{RVS} NON IN PAGINA ZERO O
     SU{DES}ROM ":GOTO1000          :rem 240
```

Piccola guida per l'input dei programmi

Molti dei programmi listati da **SUPERCOMMODORE** contengono particolari caratteri di controllo (controllo cursore, tasti colore, video-inverse, ecc.).
Per rendere chiaro ciò che dev'essere battuto quando è necessario inserire uno dei suddetti caratteri sono state stabilite le seguenti convenzioni.
Generalmente i listati per VIC 20 e C64 contengono delle «parole» racchiuse tra parentesi graffe { }; tali parole rappresentano particolari caratteri di controllo: {**GIÙ**} significa premere il simbolo del cursore verso il basso. {**5 SPAZI**} vuol dire battere 5 volte la barra-spazio.
Per indicare che un tasto dev'essere «shiftato», cioè premuto insieme al tasto SHIFT, nel listato il simbolo di questo tasto sarà sottolineato.
Per esempio S significa che dev'essere premuto il tasto S mentre è abbassato il tasto SHIFT.
La manovra farà apparire sullo schermo un piccolo «cuore».
Trovando un simbolo sottolineato chiuso tra parentesi graffe (esempio {**10 N**}) dovrà essere interpretato come «premere il carattere indicato per il numero di volte che lo precede nelle parentesi»: nel nostro esempio premere 10 volte il tasto N «shiftato».
Se il tasto da premere è indicato tra le parentesi [‹›], vuol dire che lo stesso dovrà essere premuto mentre è premuto il tasto «Commodore» (il tasto «Commodore» è quello posto nell'angolo in basso a sinistra).
Se il tasto scritto tra [‹›] è preceduto da un numero, ciò significa che il tasto dev'essere premuto per il numero di volte indicato.

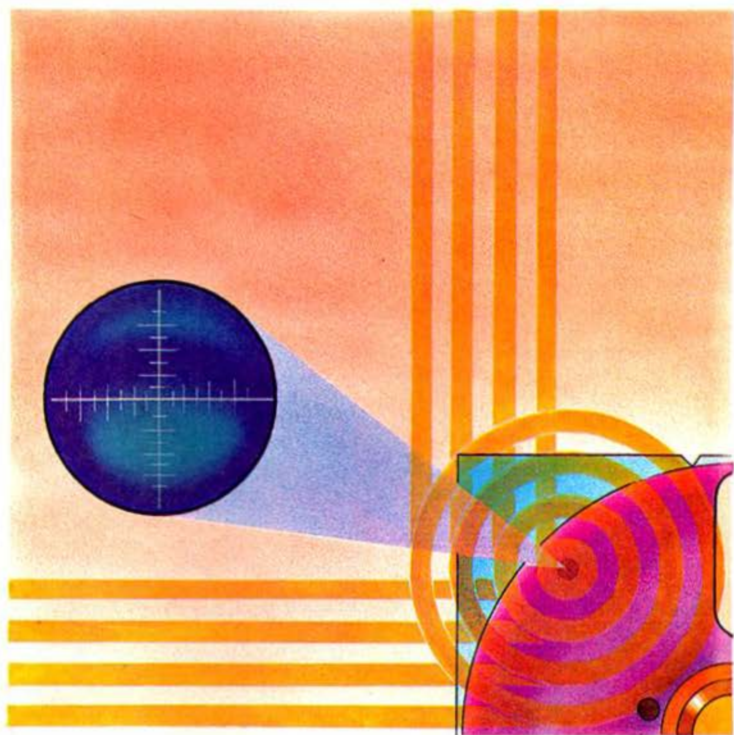
Raramente si potrà trovare un carattere alfabetico solitario racchiuso tra parentesi graffe. Tale carattere sul C64 può essere battuto mentre è premuto il tasto CTRL.
Ad esempio {**A**} sta ad indicare la sequenza CTRL-A.
A proposito del «modo virgolette» è nota la possibilità di muovere il cursore sullo schermo con i tasti CRSR.
Ogni tanto i programmatori desiderano muovere il cursore durante l'esecuzione del programma. È per questo motivo che nei programmi si troveranno dei {**SIN**}, {**HOME**} e {**BLU**}.
L'unico modo perché il calcolatore distingua il comando diretto da quello inserito nel programma è il «modo virgolette». Premendo il tasto «virgolette» (tasti SHIFT 2) il calcolatore si predispone appunto in «modo virgolette».
Battendo un carattere qualsiasi, cercando poi di correggerlo muovendo il cursore, si otterrà solo un tratto verticale in un quadratino inverso. Questo è infatti il simbolo del cursore a sinistra nel «modo virgolette».
L'unico comando di editing non utilizzabile all'interno di un programma è il DEL. Battendo nuovamente le «virgolette» il calcolatore lascia il «modo virgolette». Si può accedere al «modo virgolette» quando inserite spazi nella linea.
In ogni caso, il sistema più semplice per uscire dal «modo virgolette» è quello di premere il tasto RETURN.
Utilizzare la tabella che segue quando è necessario inserire comandi relativi al cursore e ai tasti colore.

Quando leggete:	Premete:	Vedrete:	Quando leggete:	Premete:	Vedrete:	Quando leggete:	Premete:	Vedrete:
{CLR}	SHIFT CLR/HOME		{CYN}	CTRL 4		[<7>]	C= 7	
{HOME}	CLR/HOME		{PUR}	CTRL 5		[<8>]	C= 8	
{SU}	SHIFT ⇑CRSR⇓		{GRN}	CTRL 6		{F1}	F1	
{GIU'}	⇑CRSR⇓		{BLU}	CTRL 7		{F2}	F2	
{SIN}	SHIFT ⇐CRSR⇒		{YEL}	CTRL 8		{F3}	F3	
{DES}	⇐CRSR⇒		[<1>]	C= 1		{F4}	F4	
{RVS}	CTRL 9		[<2>]	C= 2		{F5}	F5	
{OFF}	CTRL 0		[<3>]	C= 3		{F6}	F6	
{BLK}	CTRL 1		[<4>]	C= 4		{F7}	F7	
{WHT}	CTRL 2		[<5>]	C= 5		{F8}	F8	
{RED}	CTRL 3		[<6>]	C= 6				

File relativi: semplicità ed economia

di **G. W. Miller** e **T. Heimarck**
trad. ed adatt.
di **M. Cristuib Grizzi**

Tra i possessori di disk drive uno degli argomenti senz'altro più confusi è come utilizzare i file relativi. Con questo articolo cercheremo di spiegare come essi funzionano e come crearsi un piccolo archivio che li utilizzi.



Possedete un disk drive 1541? Se è così, disponete di ben 163 Kbyte di memoria "extra" per le variabili. Precisamente, 167132 byte di RAM aggiuntiva per il vostro C64, VIC 20, C16 o Plus/4, che diventano accessibili con l'uso dei file relativi.
Il drive 1541 immagazzina le informazioni in diversi tipi di file, il più comune dei quali è il file programma (PRG). Se si scrive un programma BASIC e lo si salva quindi su disco con SAVE"NOME",8, esso viene immagazzinato sul disco come file programma.
I file sequenziali sono anch'essi di semplice utilizzo: come i PRG, contengono dati, anche se in forma diversa, e vengono creati generalmente con il comando PRINT #.
L'uso dei file relativi aggiunge solo un po' più di linee al vostro programma di archivio che utilizzi i sequenziali, e non è di difficile attuazione.
Tuttavia, quasi tutti coloro che hanno cercato di imparare ad usare i file relativi hanno incontrato una gran varietà di ostacoli, alquanto frustranti. In effetti, non esistono molte informazioni scritte sul loro principio di funzionamento, e le istruzioni date dal manuale del drive sono incomplete e a volte addirittura sbagliate.
Dopo molti tentativi infruttuosi per far funzionare correttamente un file relativo la maggioranza delle persone lascia perdere. Ciò è comprensibile, dal momento che anche un programma di utilizzo di questi file, che funzioni correttamente, può a volte visualizzare degli errori del DOS, anche se in realtà non si è verificato alcun errore.
Iniziamo dunque un viaggio – passo dopo passo – all'interno del procedimento per creare ed usare i file relativi. Sarà sorprendente accorgersi che essi non sono così difficili da usare quanto sarebbe potuto sembrare.

Trovare un libro nella biblioteca

Per prima cosa occorre capire bene le differenze tra i concetti di file, record, campo e carattere.
Un file contiene uno o più record, costituiti a loro volta da uno o più campi.
I campi sono infine costituiti da caratteri.
Immaginate di dover catalogare una piccola biblioteca di 500 libri: se non avete il computer, acquisterete un archivio con 500 schede di carta. Ogni scheda avrà lo spazio per il nome dell'autore, il titolo del libro, l'anno di pubblicazione e il codice del libro.
L'intera collezione di schede è chiamata

"file". Ogni scheda, sia essa vuota o compilata, è un "record". Ogni categoria nella scheda (autore, titolo, anno e codice) è chiamata "campo". Si noti che i record devono avere tutti la stessa misura, mentre i campi possono avere una lunghezza qualsiasi (l'autore pouò essere sia Fo – 2 caratteri – che Pirandello – 10 caratteri – e così via...).
È quindi del tutto intuitivo vedere come un campo contenga dei caratteri.
Naturalmente, ci sarà sulle schede uno spazio limitato per scrivere nome dell'autore, titolo, ecc.: ci sarà dunque un numero massimo di caratteri disponibili per ogni campo, al di sopra del quale occorrerà ricorrere ad abbreviazioni. Poniamo che la somma dei caratteri disponibili per il nome dell'autore, più quelli per il titolo del libro, più quelli per la data e il codice sia 72: sarà questa la massima capacità della scheda, e quindi quella del record.
Dal momento che abbiamo 500 libri, e quindi disponiamo di 500 schede, se viene acquistato un nuovo libro, non avremo più schede per catalogarlo. A questo fine acquisteremo una decina di schede nuove, da tenere per ogni evenienza di futuri acquisti. Il contenitore dello schedario ha però una capacità massima – poniamo – di 1200 schede: quando arriveremo a possedere il 1201 esimo libro dovremo spezzare in due l'archivio.

Perché usare i file relativi?

I file sequenziali sono facili sia da capire che da gestire all'interno di un programma: molti programmatori infatti ne fanno uso ogni qual volta occorra immagazzinare informazioni su disco dall'interno di un programma. Dal momento che i file sequenziali sono così facili, perché dovremmo romperci la testa con i file relativi?
Ritorniamo per un momento alla biblioteca. Un cliente ci ha chiesto un libro di Faulkner: se il file (cioè l'archivio) fosse sequenziale, dovremmo consultare le schede in ordine, iniziando dalla prima, dalla A alla Z. Certamente noi umani andremmo "a naso" ad iniziare la ricerca, dove possiamo supporre che ci sia la F, ma il computer non ha "naso", ed inizierebbe in ogni caso la ricerca dalla lettera A. Pensate al tempo che richiederebbe, in questo modo, la ricerca di un libro di Zoroastro...
Ci sono due buone ragioni per immagazzinare le informazioni nei file relativi: velocità ed economia di memoria. Un file relativo permette infatti un accesso molto veloce ai record individuali.
Con un file relativo si può andare direttamente nella locazione dove è immagazzinata l'informazione ed ottenere solo e proprio quell'informazione che stiamo cercando. È un po' come se il computer diventasse dotato di quel "naso" di cui parlavamo prima, ed anche di un'ulteriore precisione, dal momento che l'uomo inizierebbe a cercare *più o meno* nella posizione in cui ci dovrebbe essere la lettera F, mentre il computer andrà a colpo sicuro ad estrarre la scheda di Faulkner. Una delle ragioni più importanti per l'utilizzo dei file relativi è il fatto che essi non occupano alcuna quantità della memoria del computer, a parte quella occupata da un record individuale.

CAMPO	ESTENSIONE DEI CAMPI E LORO CONTENUTO														
NOME	A	N	T	O	N	I	O								
SECONDO NOME	M	A	X												
COGNOME	B	R	A	M	B	I	L	L	L	A					
VIA	R	E	S	T	E	L	L	I		5					
CITTÀ	M	I	L	A	N	O					STATO			CAP	
				TELEFONO							I	T	A	2	0
CAP (continua)	1	2	4	0	2	-	6	8	8	8	2	2	8		

Ogni quadratino rappresenta un byte di informazione. Le lineette in neretto mostrano l'inizio e la fine di ogni campo. I campi sono riempiti completamente quando il file viene scritto; i byte "vuoti" vengono riempiti con degli spazi, come nelle linee 210-280 del programma esemplificativo.

Con l'appropriato programma in memoria un VIC *inespanso* può gestire un file contenente ben 163 Kbyte di informazioni: invece di preoccuparvi che il programma di gestione non occupi troppo spazio in memoria, potrete essere liberi di scrivere programmi eleganti ed elaborati con decine di opzioni diverse.

Un po' di regole

I programmatori iniziano generalmente a contare da zero: la locazione di memoria più bassa è la 0, e si trova in pagina zero. I file relativi non seguono questa convenzione: il primo record è il numero uno, ed il primo carattere nel record è chiamato carattere numero uno.
Inoltre, se avete su disco due o più file relativi, potete averne uno soltanto aperto in un dato momento, anche se è possibile usare contemporaneamente file relativi e sequenziali.
I record possono avere una lunghezza massima di 254 byte: se desiderate dei record più lunghi, dovrete spezzarli in due. Ad esempio, un record di 400 caratteri può essere spezzato in due record da 200 caratteri ognuno. La prima metà dei record spezzati sarà numerata con numerazione dispari (1, 3, 5, ...), mentre la seconda metà avrà numeri pari (2, 4, 6, ...).
Il massimo numero di record che è possibile creare è 65535, anche se in pratica non si raggiungerà mai questo limite, per le limitazioni di spazio disponibile sul disco.
Quando un disco viene formattato la directory mostra 664 blocchi liberi; i blocchi, chiamati anche settori, sono aree del disco che possono contenere 256 caratteri ognuno. Due di essi vengono usati dal DOS (Disk Operating System), lasciando liberi in realtà solo 254 byte per blocco: è questa la ragione per cui la massima lunghezza di un record è 254 caratteri.
I file relativi usano speciali settori, chiamati "side sectors", per tenere il conto di quali settori contengono dati e quali invece no. Non importa ora conoscere come funziona questo procedimento, basti sapere che ogni file relativo può avere fino a sei "side sectors". Ognuno di essi può controllare fino a 120 settori (non si confondano i settori con i record!), per un totale quindi di 720 settori, numero superiore a quello dei blocchi liberi in un disco vuoto e formattato.
Riempire completamente un disco con un file relativo significa usare sei blocchi per i "side sectors", lasciando 664 – 6 = 658 settori liberi per i dati.
Quindi, 658 settori di 254 byte ognuno ci lasciano spazio per 167132 caratteri su ogni disco. Questa è la massima capacità di un file relativo su disco.
Ciò significa che si può riempire completamente un disco con 658 record di 254 caratteri ciascuno, oppure 1671 record di 100 caratteri ciascuno, o qualsiasi altra combinazione di numero di record e numero di caratteri per record che, moltiplicati tra di loro, diano come risultato un numero minore od uguale a 167132.

Come gestire i file relativi

In un file relativo ogni record è numerato e tutti i record devono essere contraddistinti dalla medesima lunghezza. Normalmente, i campi all'interno di un record hanno anch'essi una lunghezza predeterminata.
Per trovare un record all'interno del file tutto ciò che occorre sapere è il suo numero. Ciò può suonare come un problema, ma vedremo qualche trucchetto per semplificare il procedimento di ricerca.

Sfortunatamente, la Commodore non ha incluso nel BASIC del C64 (e nemmeno in quello dei C16-Plus/4) alcun comando diretto per la gestione dei file relativi, al contrario invece di quanto accade con il BASIC 7.0 del C128.
Possiamo comunque creare e manipolare i file relativi usando i familiari comandi della gestione dei sequenziali, più due comandi per il drive che appariranno meno familiari.

Il primo passo

Iniziamo a creare un file relativo. Occorrono tre operazioni:

1. aprire il file e stabilire la lunghezza dei record;
2. contrassegnare l'ultimo record del file;
3. chiudere il file.

OPEN è il comando usato per aprire qualsiasi tipo di file, compreso quindi un file relativo. La forma del comando è solo leggermente diversa da quella usata per aprire un canale con la stampante, od aprire un file sequenziale su disco:

```
OPEN1,8,2, "NOME FILE,L," + CHR$
(LUNGH.RECORD)
```

L'istruzione inizia come qualsiasi altro comando OPEN: 1 è il numero del file logico usato, 8 è il numero di periferica del drive, 2 è il numero del canale di comunicazione usato (può essere indifferentemente un numero tra 2 e 14, ma *ricordate quale numero di canale avete usato, perché ne avrete bisogno più avanti*).
Segue una virgola ed il nome del file tra virgolette. Qui arriva la prima novità: al nome del file, sempre all'interno delle virgolette, segue una virgola, la lettera L e un'altra virgola. A questo punto si chiudono le virgolette. La lettera L sta per Lenght, lunghezza dei record. Questo comando dice al DOS che deve essere aperto un file relativo, la cui lunghezza dei record è contenuta nell'istruzione CHR$. Si ricordi che la massima lunghezza ammessa è 254.
Definire la lunghezza dei record è assolutamente necessario quando si crea un file relativo. Nel momento in cui dovremo eventualmente espandere il file con l'aggiunta di nuovi record, sarà ancora necessario specificarne la lunghezza.
Se si lavora con un file relativo già presente sul disco, si può invece usare un semplice comando OPEN:

```
OPEN1,8,2,"NOME FILE"
```

Non occorre in questo caso dire al drive che si tratta di un file relativo; esso se ne accorge automaticamente quando lo trova sul disco. Non occorre nemmeno fornire la lunghezza dei record, che allo stesso modo viene automaticamente "letta" dal drive. Infine non occorre nemmeno specificare se il file è stato aperto per la scrittura o per la lettura, in quanto si possono eseguire entrambe le operazioni.
Il passo seguente aiuta a risparmiare un po' di tempo quando si usa il file: decidiamo il numero di record con il quale – per il momento – creare il nostro file. Potremo aggiungere dei record quando vorremo, più avanti. Dato che il DOS richiede un certo tempo per creare il file sul disco, conviene aprirlo e creare i recod *prima* di immagazzinare qualsiasi dato.
Ritornando all'esempio dell'archivio per la biblioteca, è molto più conveniente comprare un certo numero di schede vuote tutte in una volta, piuttosto che perdere tempo per andare in cartoleria ad acquistare una nuova scheda per ogni libro che decidiamo di catalogare.

Posizionare un puntatore

C'è un altro comando che bisogna imparare; ha questa forma:

```
OPEN15,8,15
PRINT #15, "P" + CHR$(nc+96) +
CHR.DOL$(bb) + CHR$(ba) +
CHR.dol$(pl)
```

Il canale 15 è quello dei comandi per il drive. Dobbiamo in questo caso aprirlo ed inviare al drive sei caratteri. Per primo la lettera P (per Posizione, o Puntatore). Questa P dice al drive di cercare un certo record. Il secondo carattere è il numero di canale (nc) sommato a 96: se il file era stato aperto con OPEN1,8,2, nc vale 2.
I due numeri seguenti specificano un certo record come somma del byte basso più 256 volte il byte alto. Il record numero 300, ad esempio, si tradurrà in CHR$(44)+CHR$(1), dal momento che 44+(1*256)=300. Ricordiamo che il byte basso (bb) ed il byte alto (ba) si calcolano trasformando il numero del record in un numero esadecimale di quattro cifre (300 diventa 012C), spezzandolo in due (01 e 2C) e riconvertendo i due numeri ottenuti in decimale (01 in decimale è 1= byte alto, mentre 2C in decimale è 44= byte basso). Sembra molto complicato, ma non preoccupatevi: tutti questi calcoli non dovrete farli voi, ma toccherà al programma di gestione file calcolare questi parametri.
L'ultimo carattere (pl) indica in quale parte del record occorre iniziare a leggere o scrivere. In molti casi sarà un CHR$(1), per indicare che vogliamo iniziare dall'inizio del record, al primo carattere. Per iniziare dal sesto carattere, ad esempio, useremo un CHR.$(6).
In generale occorre aprire dapprima il canale 15, quindi aprire il file relativo e posizionare il puntatore. Leggere e scrivere sul file, se è necessario, quindi chiudere sia il file relativo che il canale di comando.
OPEN15 è la prima operazione da compiere, mentre CLOSE15 sarà l'ultima.

Il programma

Con propositi dimostrativi analizzeremo un semplice programma di gestione dei file relativi. Per le sue caratteristiche didattiche, il programma non presenta un'utilità pratica; è tuttavia certamente possibile sfruttare le tecniche presentate per scrivere un database, che utilizzi i file relativi per qualsiasi tipo di archivio desiderato; modificate a piacere le linee del programma a seconda delle vostre esigenze, magari aggiungendo delle routine di controllo degli errori di input, od altre migliorie che potreste ritenere necessarie.
È importante ricordare che la sintassi delle istruzioni per la gestione del file *deve* essere assolutamente esatta. Qualunque deviazione, anche piccola, causerà errori che potranno risultare di molto difficile individuazione.
Per prima cosa occorre decidere che tipo di informazioni vogliamo immagazzinare. Per un archivio indirizzi – ad esempio – dovremo inserire nome, via, città, stato, codice postale ed eventualmente il numero di telefono.
Ci semplifica di molto le cose il porre la lunghezza dei campi uguale per tutti i record. Decidiamo il massimo numero di caratteri che verrà immagazzinato nel campo del nome di battesimo, dell'eventuale secondo nome, del cognome e così via.
È molto utile stabilire questi parametri dapprima su un foglio di carta. Ecco come possiamo definire i campi per l'archivio indirizzi:

Nome di battesimo	15 caratteri
Secondo nome	15 caratteri
Cognome	15 caratteri
Via	30 caratteri
Città	25 caratteri
Stato (abbreviato)	3 caratteri
Codice postale	5 caratteri
Numero di telefono	12 caratteri
Fine del record	1carattere
TOTALE	121 caratteri

Ora che abbiamo determinato che la lunghezza totale dei record sarà di 121 caratteri siamo pronti per cominciare.
Per risparmiare un po' di tempo nell'esecuzione del programma decidiamo subito con quale lunghezza del file iniziare (cioè il numero dei record per ora vuoti). Ricordiamo che potremo in qualsiasi momento aumentare questa lunghezza senza dover ricominciare tutto da capo, ma semplicemente scrivendo dei record contraddistinti da un numero superiore all'attuale fine del file.
Siamo pronti per scrivere la prima parte del programma per file relativi.
Quando si manderà in esecuzione il programma risponderemo alla domanda di linea 10 con 121, la lunghezza del record che abbiamo precedentemente stabilito.
Alla domanda di linea 20 risponderemo

con 50. Naturalmente, il valore è a scelta, con l'unica limitazione dello spazio disponibile sul disco (50 record di 121 caratteri usano poco più di 6 Kbyte di spazio sul disco, cioè circa 25 blocchi).

La cosa migliore è usare un disco apposito per l'immagazzinamento dati di un file relativo, separato dal disco su cui è registrato il programma di archivio.

Per posizionare il file sul record dobbiamo aprire il canale di comando e dire al DOS cosa intendiamo fare.

La linea 50 apre il canale, la linea 60 apre un file relativo di lunghezza RL e la linea 70 dice al DOS di posizionare la testina di lettura/scrittura del drive sul record appropriato. Il numero del record viene trasmesso nel formato byte basso/byte alto, che è determinato dalle linee 30 e 40.

Infine scriveremo la parola END sull'ultimo record. Ciò libera spazio per il file e pone il valore 255 nel primo byte di ogni record sul disco. Questo processo può richiedere diversi minuti, specialmente se il file che si apre è molto lungo.

A questo punto il file su disco è stato creato, e può essere chiuso con la linea 90.

Un errore inesistente

La cosa più logica, a questo punto, sarebbe chiudere il canale di comando (CLOSE15), ma se lo facessimo, la luce rossa del drive inizierebbe a lampeggiare, segnalando un errore. In realtà non si è verificato alcun errore: ciò che accade in questa situazione è che abbiamo scritto su un record contraddistinto da un numero più alto di quello che il DOS si aspetta, ed esso invia quindi un messaggio d'errore RECORD NOT PRESENT. La linea 95 legge il canale degli errori e visualizza il "falso" messaggio d'errore. Non è il caso di preoccuparsene: il suo significato è semplicemente di avvertire che questo record non esisteva precedentemente sul disco.

Abbiamo ora un file relativo su disco pronto per ricevere dati e possiamo iniziare a preparare questi ultimi per il loro immagazzinamento.

Preparare il record

Il nome "Max" contiene solo tre caratteri, mentre il nome "Antonio" ne contiene sette. Sarà conveniente far iniziare allo stesso punto del record sia il cognome di Max che quello di Antonio; per ottenere questo scopo possiamo concatenare il nome al cognome, tramite l'aggiunta di un numero opportuno di spazi vuoti.

Le linee 210-280 eseguono questa operazione, in modo che la stringa RC$ contenga tutti dati, con il giusto numero di caratteri per ogni campo.

Scrivere su disco

Per scrivere dati sul disco si segue lo stesso schema usato nella creazione del

1. apertura del file;
2. posizionamento del puntatore sul record che si intende scrivere;
3. scrittura del record;
4. chiusura del file (opzionale).

La linea 350 scrive le informazioni sul disco, la 370 legge il canale degli errori (se tutto va bene, deve visualizzare OK) e la linea 380 ci chiede l'inserimento successivo.

Avrete notato che non abbiamo cercato di inviare il separatore di fine record cui si accennava precedentemente. Sembrerebbe invece che abbiamo inserito in ogni record (che può contenere 121 caratteri) solo 120 caratteri. In realtà ne abbiamo inseriti 121: il comando PRINT# – proprio come PRINT – aggiunge un RETURN (CHR$(13)) in modo automatico, se la stringa non è seguita da una virgola o un punto e virgola. C'è quindi un 13 nella posizione 121, ed è il separatore di fine record.

È una buona idea riempire con informazioni l'intero record, anche se i campi vuoti vengono semplicemente definiti come spazi. Se non si segue questa procedura, il record conterrà semplicemente dei CHR$(0).

Leggere il file

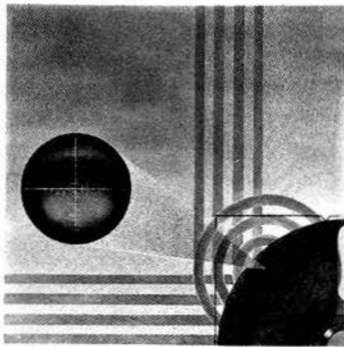
Dopo aver scritto il file ci viene chiesto se vogliamo effettuare altri inserimenti. Rispondendo negativamente, passeremo alla fase di lettura dei dati.

Questa si svolge nel modo seguente:

1. apertura del file;
2. posizionamento del puntatore sul record che si desidera leggere;
3. lettura del record;
4. chiusura del file (opzionale).

Una volta che i dati sono stati immagazzinati su disco la loro lettura è molto facile. Se il record contiene meno di 80 caratteri, si può usare il comando INPUT#, ma dal momento che abbiamo usato record con lunghezza di 120 caratteri, dovremo usare l'istruzione GET# e concatenare le stringhe per costruire la stringa finale RC$, operazione compiuta dalla linea 560.

Le linee 500-560 leggeranno i record



dal disco, in sequenza, dal numero più basso al numero più alto. I passi per ottenere questo risultato dovrebbero ora apparire familiari: apertura del canale 15, apertura del file relativo, posizionamento del puntatore e trasferimento dei caratteri dal disco alla RAM, attraverso l'uso del comando GET#.

Le linee 699-810 permettono di ottenere la miglior caratteristica dei file relativi: la lettura diretta di un qualunque record all'interno del file, semplicemente specificandone il numero. Il programma, oltre a chiedere il numero del record che si vuole leggere chiede anche da quale posizione – all'interno del record – si vuole iniziare la lettura. Per evitare, ad esempio, la visualizzazione dei nomi inserite 16. Per visualizzare solo i numeri di telefono inserite 109.

Modificare un record

La possibilità di poter modificare un singolo record senza dover riscrivere tutto il file è una delle caratteristiche che rendono così potente l'uso dei file relativi.

Anche qui si inizia con l'aprire il canale di comando ed il file, posizionare il puntatore sul record e quindi con il leggere in memoria l'intero record. A questo punto è possibile effettuare tutte le modifiche desiderate a concatenare tutti i campi all'interno del record, come visto precedentemente.

Quindi si riposiziona il file usando:

PRINT#15, "P" + CHR$ (2) + CHR$ (bb) + CHR$ (ba) + CHR$ (primo byte)

Ora si può usare PRINT# per riscrivere il record. Infine *occorre posizionare il puntatore ancora sul primo carattere del record.* Ciò potrà sembrare superfluo, dal momento che ci siamo posizionati sul record prima di scrivere i dati, ma se non si segue correttamente questa procedura, i nostri dati potrebbero essere alterati. Per maggiori dettagli su questo problema vedi nell'ultima parte dell'articolo il paragrafo "La controversia sui file relativi".

L'ultimo passo

Con i file relativi può essere immagazzinata su disco una gran quantità di dati, tuttavia essi non saranno di alcuna utilità, se non siamo in grado di utilizzarli correttamente.

Possiamo caricare in memoria ogni record in qualunque momento, ma con la lentezza del drive 1541 il procedimento può diventare piuttosto lungo e noioso, specialmente se si usano file molto lunghi. Addio ai sogni di grande velocità, che hanno fatto nascere l'idea di utilizzare i file relativi...

Esistono fortunatamente delle tecniche che permettono di velocizzare tutto il procedimento. Il modo più semplice per gestire una ricerca all'interno di un file relativo è creare dei cosiddetti file "indice" per i campi chiave che si desiderano ricercare.

Questi file "indice" sono generalmente

dei sequenziali già in ordine alfabetico – o comunque alfabetizzabili con opportune routine di sort – e caricati in una matrice appositamente creata da un'istruzione DIM. Non è necessario leggere il file "indice" ogni volta che si opera sul file relativo, ma solo quando si desidera ricercare velocemente un dato record e non se ne conosca il numero progressivo. Il file "indice", come avrete già intuito, assocerà il contenuto del campo chiave al numero del record nel file relativo, permettendone una ricerca immediata.

La controversia sui file relativi: idee sbagliate e disinformazione

La letteratura esistente sui file relativi è quanto mai infarcita di inesattezze e di confusione. Il manuale del drive e vari libri dedicati all'argomento contengono errori ed informazioni sbagliate su come creare i file relativi e sul loro principio di funzionamento. Passiamo in rassegna alcune delle credenze più diffuse circa questo tipo di file ed analizziamo la loro attendibilità:

CI SONO NUMEROSI "BUG" ALL'INTERNO DEL DOS PER QUANTO RIGUARDA I FILE RELATIVI

Falso, anche se soltanto in parte. Esiste un unico "bug" che influenza il trattamento dei file relativi, e viene descritto più avanti nell'articolo . Esso è comunque raro e può essere evitato molto facilmente. Inoltre i file relativi provocano a volte messaggi di errore "falsi" da parte del DOS che hanno tuttavia una ragion d'essere, se correttamente interpretati.
Quando si lavora con i file relativi, se ne ricordino i limiti: un massimo di 254 caratteri per record, un massimo di 65535 record per file e un massimo di 167132 caratteri per disco.

CON I FILE RELATIVI SI POSSONO AVERE AL MASSIMO 720 RECORD PER FILE

Falso. Questo è l'errore più comune contenuto nei libri dedicati all'argomento, e lo stesso manuale del drive lascia intuire questa limitazione.
I file relativi sono veloci per il loro utilizzo dei "side sectors". Quando si chiede – ad esempio – di leggere il record 533, il DOS usa i "side sectors" per stabilire dove si trova il record 533 all'interno del disco, quindi salta direttamente su quel settore e ne legge il contenuto.
Ognuno dei sei "side sectors" può gestire fino a 120 diversi settori, per un totale di 720: più che abbastanza per coprire il massimo di 658 settori disponibili per i file relativi su un disco nuovo e formattato. Alcuni autori hanno interpretato questo rapporto tra "side sectors" e numero di settori per arguire un numero massimo di 720 record.
In realtà 720 è il massimo numero di settori che possono essere indirizzati dai "side sectors", ed è un limite mai raggiungibile, dal momento che un disco completamente vuoto contiene soltanto 664 settori, sei dei quali vengono inoltre riservati per gli stessi "side sectors".
Il massimo numero di record è quindi 65535 (anche se, con un numero così alto di record, ognuno di essi occuperebbe solo due byte per poter entrare nel limite dello spazio disponibile sul disco).

CON I FILE RELATIVI BISOGNA IGNORARE I MESSAGGI DI ERRORE DEL DOS

Falso. Con una sola eccezione, potete invece ignorare tutte le raccomandazioni che vi dicono di ignorare i messaggi d'errore. In genere, tre errori sono comunemente associati alla lettura e alla scrittura dei file relativi:

50 RECORD NOT PRESENT
51 OVERFLOW IN RECORD
52 FILE TOO LARGE

L'errore 50 indica che avete cercato di accedere (leggere o scrivere) ad un record il cui numero è superiore al numero di record presenti sul disco. Se state creando un nuovo file, oppure espandendo un file già esistente, potete ignorare questo errore (infatti, se scrivete un nuovo record nel file, accedete a un record che non esisteva precedentemente). *Un errore 50 conferma di fatto che la creazione o l'espansione del file è avvenuta con pieno successo.* Potete eliminare il lampeggiamento della spia rossa del drive leggendo il canale degli errori (INPUT # 15,E,E$): E deve valere 50, ed E$ deve essere "RECORD NOT PRESENT".
L'errore 51 indica che avete cercato di inviare al disco un numero di caratteri superiore a quello su cui era stato inizialmente "formattato" il file, ad esempio se cercate di scrivere 122 caratteri in un record creato per contenerne 121.
L'errore 52 deriva dal tentativo di creare dei file troppo grandi per poter essere contenuti nello spazio libero rimanente sul disco. Se su quest'ultimo sono contenuti altri programmi o comunque altri file, diminuisce logicamente lo spazio per i file relativi: una buona ragione per dedicare un intero disco a ciascun file relativo.
Quando si immagazzinano dati in file sequenziali o relativi, è una buona idea leggere regolarmente il canale degli errori. Se create un nuovo file, controllate l'errore 50; in altri casi un errore del disco indica qualche problema con il file. Soprattutto, non ignorate i messaggi d'errore.

Come creare i nuovi file

In generale occorre creare un file vuoto prima di iniziare ad utilizzarlo per i dati. Se pensate di avere 60 record, alcuni libri raccomandano di scrivere subito qualcosa in tutti i 60 record. Ciò non è realmente necessario: tutto ciò che dovete fare è invece scrivere qualcosa nel record 60: gli altri 59 verranno creati automaticamente, ed ognuno dei record vuoti conterrà un CHR$(255), seguito da una serie di CHR$ (0).
Quando viene creato un file relativo occorre definire, come abbiamo visto, lunghezza dei record con ",L,"+CHR$ (lunghezza dei record). Dopo questa operazione specificare la lunghezza dei record è opzionale (a meno che stiate espandendo il file, nel qual caso è obbligatoria). Certamente non guasta aggiungere questa lunghezza nelle istruzioni OPEN, ma rimane comunque un'operazione che è possibile evitare.

BISOGNA LASCIARE SEMPRE LO SPAZIO PER UN CHR$(13)

Vero. Se desiderate 15 caratteri per record, dovete definirne una lunghezza pari a 16. L'istruzione PRINT # aggiunge sempre un CHR$(13) alle stringhe che non terminano con una virgola o un punto e virgola.
Che succede se non si aggiunge 1 alla lunghezza dei record? Inviare 15 caratteri più un CHR$(13) equivale all'inviare 16 caratteri: se riempite con 16 caratteri un record la cui lunghezza è stata definita di 15 caratteri, otterrete un errore 51, OVERFLOW IN RECORD. La luce rossa del drive lampeggerà ogni volta che tenterete di scrivere un record.

I pro e i contro dei separatori di campo

Esistono due modi per immaggazzinare e per richiamare i dati da un file relativo.
Il primo metodo consiste nel concatenare i campi ed inviare al record una luna ed unica stringa.

```
RC$=A$+B$+C$+D$:PRINT # 1,RC$
```

Per rileggere il dato in memoria usate GET # (per ottenere un carattere alla volta), quindi la funzione MID$ per spezzare il record nei diversi campi.
L'altro metodo consiste nel separare i campi con un CHR$(13), oppure una virgola:

```
W$=CHR$(13) : RC$=A$ + W$ + B$
+ W$ + C$ + D$ : PRINT # 1, RC$
```

Questa volta, invece dell'istruzione GET # all'interno di un ciclo FOR/NEXT, si può regolare il puntatore ed usare l'ustrizone:

```
INPUT # 1, A$, B$, C$, D$
```

Il vantaggio è che INPUT # è leggermente più veloce di GET #. È anche possibile creare campi di lunghezza variabile: non occorrerà in questo caso riempire con spazi i caratteri non occupati dei campi. Lo svantaggio risiede tutta-

via nel fatto che non sarà più possibile usare né virgole né CHR$(13) all'interno di un campo. Inoltre la lunghezza dei record dovrà comunque essere leggermente superiore, per via dei separatori.

Un "bug" nei file relativi

L'istruzione INPUT # può a volte commettere errori nel leggere tutti i caratteri di un file relativo, quando si aggiornano record in modo sequenziale. Si può eliminare questo "bug" con la semplice precauzione di posizionare il puntatore al record *prima* e *dopo* la scrittura del record stesso.

```
540 PRINT#15,"P"+CHR$(2+96)+CHR$
    (103)+CHR$(2)+CHR$(1)
550 PRINT#1,"NUOVI DATI"
560 PRINT#15,"P"+CHR$(2+96)+CHR$
    (103)+CHR$(2)+CHR$(1)
```

Se il puntatore viene posizionato prima e dopo la scrittura, non avrete mai a che fare con questo "bug".
Esso è da mettere in relazione a quanto segue.
Ogni settore del disco contiene 256 byte, numerati da 0 a 255. I primi due byte vengono usati dal DOS, lasciandone 254 disponibili per programmi e file dati.
Un file relativo contenente 200 caratteri per record occupa i settori del disco come segue:

SETTORE	BYTE	NUMERO RECORD
1	0-1	usato dal DOS
1	2-201	record 1
1	202-255	record 2 (prima parte)
2	0-1	usato dal DOS
2	2-147	record 2 (resto)
2	148-255	record 3 (prima parte)
3	0-1	usato dal DOS
3	2-95	record 3 (resto)

Il record 1 è immagazzinato in prossimità dell'inizio del settore 1, e viene internamente contenuto in quel settore, mentre i record 2 e 3 sono "spezzati" in due diversi settori. Il "bug" si presenta se sono verificate le seguenti condizioni:
1. si sta aggiornando sequenzialmente il file, usando cicli FOR/NEXT per leggere record individuali, modificarne il contenuto e quindi riscriverli sul disco;
2. si scrivono soltanto pochi caratteri su un record che è immagazzinato a partire dall'inizio di un settore (nell'esempio, se si scrivono meno di 199 carattere nel record 1);
3. il numero di caratteri inserito in questo record è minore del numero di caratteri che costituiscono la continuazione di altri record occupanti più di un settore (i record 2 e 3 dell'esempio).
Il problema non si ripresenterà fino a quando si abbia a che fare con un record contenuto all'inizio di un settore di numero pari.
Senz'altro tutto ciò suonerà molto complicato, quindi passiamo alla pratica e facciamo in modo che il "bug" si riveli, attraverso un semplice esempio.
Per prima cosa creiamo un file relativo con sei record di 252 caratteri ciascuno:

```
10 OPEN15,8,15:OPEN1,8,2,"0:ABC,
   L,"+CHR$(252)
20 PRINT#15,"P"+CHR$(2+96)+CHR$(
   6)+CHR$(0)+CHR$(1)
30 PRINT#1,"ULTIMO RECORD"
40 INPUT#15,E,E$:PRINTE$:CLOSE1:
   CLOSE15
```

La linea 10 apre il canale di comando e un file relativo con lunghezza dei record pari a 252. La linea 20 posiziona il puntatore sul record numero sei. La linea 30 inserisce i caratteri "ULTIMO RECORD" nel record 6 (creando automaticamente i cinque precedenti record vuoti) e la linea 40 chiude il file.
Ora inseriremo dei dati in ciascun record. Digitate NEW e copiate questo programmino:

```
10 A$="123456789"
20 OPEN15,8,15:OPEN1,8,2,"0:ABC"
30 FORJ=1TO5:GOSUB500
40 PRINT#1,A$:NEXT
50 REM ORA CAMBIA I DATI
60 J=1:GOSUB500:PRINT#1,"A"
70 FORJ=2TO5:GOSUB500
80 INPUT#1,B$:PRINT"RECORD #";J,
   B$
90 B$=B$+STR$(J)
100 PRINT"MODIFICATO IN ";B$
110 GOSUB500:PRINT#1,B$
120 NEXT
130 FORJ=1TO5:GOSUB500:INPUT#1,C
    $:PRINTJ,C$:NEXT
140 CLOSE1:CLOSE15
499 END
500 PRINT#15,"P"+CHR$(2+96)+CHR$
    (J)+CHR$(0)+CHR$(1):RETURN
```

La soubroutine di linea 500 posiziona il puntatore sul record J.
Per prima cosa le linee 10-40 inseriscono A$ ("123456789") nei primi cinque record. La linea 60 modifica il record 1 (che è contenuto all'inizio di un settore) e fa in modo che essa contenga soltanto il singolo carattere "A" (provate con "AB" o "ABC" e vedete che cosa succede).
Nelle linee 70-120 si aggiornano i record 2-5, leggendo B$ e quindi aggiungendo a questa stringa il numero del record. La linea 80 legge il record con INPUT #, lo visualizza sullo schermo e vi aggiunge l'equivalente in stringa del numero J (STR$(J)).
Infine le linee 130-140 leggono il file e visualizzano il contenuto di ciascun record. Fate molta attenzione al record 3, il cui contenuto è stato "corrotto" dal famoso "bug".
Ora, aggiungete questa linea:

115 GOSUB500

La subroutine posiziona il puntatore ai record. Date ancora il RUN al programma e il fastidioso "bug" sarà scomparso.
Speriamo con questa trattazione di aver chiarito molti dei dubbi sul trattamento dei file relativi. Naturalmente si tratta di un argomento un po' "tecnico", che per essere affrontato, richiede dal lettore almeno la conoscenza della gestione dei file sequenziali. Quest'ultima è stata trattata nella serie di articoli "Iniziare con un sistema a dischi" sui numeri 3, 4, 5 e 6 del 1984.

File relativi per VIC 20, C64, C16 e Plus/4

```
5 REM ** CREA UN FILE RELATIVO **
                              :rem   5
10 PRINT"{CLR}LUNGHEZZA DEI RECORD":INPUT
   RL:RL=RL+1                 :rem 133
20 PRINT"NUMERO TOTALE DI RECORD":INPUTRN
   :RN=RN+1                   :rem 155
30 HB=INT(RN/256)             :rem 210
40 LB=RN-HB*256               :rem  77
50 OPEN15,8,15                :rem 243
60 OPEN1,8,2,"REL.FILE,L,"+CHR$(RL)
                              :rem 235
70 PRINT#15,"P"+CHR$(2+96)+CHR$(LB)+CHR$(
   HB)+CHR$(1)                :rem  70
80 PRINT#1,"END"              :rem 144
90 CLOSE1                     :rem  16
95 INPUT#15,E,E$:PRINT"{RVS}STATO DISCO:"
   :PRINTE$:CLOSE15           :rem  39
99 REM ** PREPARA PER LA SCRITTURA **
                              :rem  62
100 RC=RC+1                   :rem  84
110 RC$=""                    :rem 204
120 SP$="{ 30 SPAZI}"         :rem 219
130 PRINT"NOME":INPUTNF$      :rem  22
140 PRINT"SECONDO NOME":INPUTMN$:rem 41
150 PRINT"COGNOME":INPUTLN$   :rem 247
160 PRINT"VIA":INPUTSA$       :rem 202
```

```
170 PRINT"CITTA'":INPUTCI$            :rem 127
180 PRINT"STATO":INPUTST$             :rem 138
190 PRINT"CAP":INPUTZI$               :rem 208
200 PRINT"TELEFONO":INPUTPH$          :rem 69
210 RC$=RC$+LEFT$(NF$+SP$,15)         :rem 73
220 RC$=RC$+LEFT$(MN$+SP$,15)         :rem 81
230 RC$=RC$+LEFT$(LN$+SP$,15)         :rem 81
240 RC$=RC$+LEFT$(SA$+SP$,15)         :rem 76
250 RC$=RC$+LEFT$(CI$+SP$,15)         :rem 69
260 RC$=RC$+LEFT$(ST$+SP$,15)         :rem 97
270 RC$=RC$+LEFT$(ZI$+SP$,15)         :rem 94
280 RC$=RC$+LEFT$(PH$+SP$,15)         :rem 84
299 REM ** SCRIVE SUL FILE **         :rem 16
300 OPEN15,8,15                       :rem 33
310 OPEN1,8,2,"REL.FILE,L"            :rem 210
320 HB=INT(RC/256)                    :rem 249
330 LB=RC-HB*256                      :rem 116
340 PRINT#15,"P"+CHR$(2+96)+CHR$(LB)+CHR$
    (HB)+CHR$(1)                      :rem 118
350 PRINT#1,RC$                       :rem 94
360 CLOSE1                            :rem 64
370 INPUT#15,E,E$:PRINT"{RVS}STATO DISCO:
    ":PRINTE$:CLOSE15                 :rem 83
380 PRINT:PRINT"UN ALTRO INSERIMENTO?"
                                      :rem 228
390 GETG$:IFG$=""THEN390              :rem 103
400 IFG$="N"THEN500                   :rem 33
410 IFG$="S"THEN100                   :rem 35
420 GOTO390                           :rem 107
499 REM ** LEGGE SEQUENZIALMENTE IL FILE
    RELATIVO **                       :rem 43
500 FORA=1TORC:RC$="":PRINT"RECORD #";A
                                      :rem 64
510 OPEN15,8,15                       :rem 36
520 OPEN1,8,2,"REL.FILE"              :rem 93
530 HB=INT(A/256)                     :rem 168
540 LB=A-HB*256                       :rem 35
550 PRINT#15,"P"+CHR$(2+96)+CHR$(LB)+CHR$
    (HB)+CHR$(1)                      :rem 121
560 FORB=1TO120:GET#1,A$:RC$=RC$+A$:NEXT
                                      :rem 89
570 CLOSE1:CLOSE15                    :rem 89
580 PRINTMID$(RC$,1,15)               :rem 33
590 PRINTMID$(RC$,16,15)              :rem 88
600 PRINTMID$(RC$,31,15)              :rem 77
610 PRINTMID$(RC$,46,15)              :rem 84
620 PRINTMID$(RC$,76,15)              :rem 88
630 PRINTMID$(RC$,101,15)             :rem 126
640 PRINTMID$(RC$,104,15)             :rem 130
650 PRINTMID$(RC$,109,15)             :rem 136
660 NEXT                              :rem 219
699 REM ** LEGGE RECORD INDIVIDUALE **
                                      :rem 133
700 PRINT"# RECORD DA LEGGERE":INPUTRE
                                      :rem 43
710 PRINT"A PARTIRE DALLA POSIZIONE":INPU
    TPO                               :rem 72
720 HB=INT(RE/256):LB=RE-HB*256       :rem 25
730 OPEN15,8,15                       :rem 40
740 OPEN1,8,2,"REL.FILE"              :rem 97
750 PRINT#15,"P"+CHR$(2+96)+CHR$(LB)+CHR$
    (HB)+CHR$(PO)                     :rem 233
760 FORA=POTO120:GET#1,A$:PRINTA$;:NEXT
                                      :rem 182
770 INPUT#15,E,E$:PRINT"{RVS}STATO DISCO:
    ":PRINTE$                         :rem 65
780 CLOSE1:CLOSE15                    :rem 92
790 PRINT"PREMI UN TASTO PER CONTINUARE"
                                      :rem 251
800 GETG$:IFG$=""THEN800              :rem 95
810 GOTO700                           :rem 105
```

REM:HW

Hardware

Ecco il nuovo Commodore 128

di **C. Brannon**
trad. ed adatt.
di **M. Cristuib Grizzi**

Il nuovo C128 crea molto interesse, specialmente tra i possessori del diffuso C64. Ma che cosa rappresenta realmente il C128? È solo una versione del C64 a 128 Kbyte? Vediamo di scoprirne i segreti.



Poco tempo dopo la sua comparsa sul mercato il C64 si è dimostrato il leader indiscusso della nuova generazione di home computer: anche al suo prezzo iniziale di ottocentomila lire, il C64 possedeva una quantità di memoria offerta da macchine che ai tempi costavano milioni, insieme ad una grafica di ottima qualità, la possibilità di creare animazioni ed un eccezionale sintetizzatore sonoro, che ha portato un tocco di realismo a quelli che prima erano solo dei beep.
Il C64 è in breve diventato un best seller, con più di due milioni di unità vendute in tutto il mondo e circa seimila programmi disponibili di tutti i generi.
È raro, nel mondo odierno della produzione di home computer, assistere a un fenomeno come quello che ha visto il C64 rimanere tale e quale per tre anni, senza nessuna innovazione o miglioria; del resto, la sua diffusione è tale che ben pochi sarebbero disposti a rinunciare alla propria ben fornita collezione di programmi – ed alla possibilità di scambiarli con altri amici anch'essi possessori di C64 – per acquistare un nuovo modello Commodore più potente (come ad esempio il Plus/4), ma con poco software disponibile.

Dopo lo scarso successo ottenuto dal Plus/4 – forse proprio per questo motivo – la Commodore ha mutato filosofia ed ha lanciato il C128, perfettamente compatibile con il C64 e dotato di maggiore potenza e flessibilità.
Il C128 è letteralmente costituito da tre computer in uno: un C64 con il familiare video a 40 colonne, SPRITE, SID e il BASIC V2; un C64 potenziato con 128 Kbyte, tutte le caratteristiche standard del C64, video a 40/80 colonne e BASIC 7.0; un computer CP/M compatibile, che promette la possibilità di far "girare" l'incredibile quantità esistente di software sotto CP/M. Confrontata con quella del C64, la consolle del C128 è molto più grande, probabilmente per poter gestire in modo non troppo complicato l'hardware dei tre computer in uno. La parte principale della tastiera è identica a quella del C64, con la differenza che i tasti funzione sono stati spostati nell'angolo superiore destro e ridisposti orizzontalmente. È inoltre presente un tastierino numerico con i tasti "+", "–" ed ENTER (una duplicazione del tasto RETURN). Lungo la parte alta della tastiera si trovano i tasti ESC, TAB, ALT, CAPS, LOCK, HELP, LINE FEED, 40/80 DISPLAY e quattro tasti cursore separati.
Nessuno di questi tasti addizionali, com-

preso il tastierino numerico ed i tasti cursore separati, funziona in modo C64 per conservare la totale compatibilità tra le due macchine. Sarebbe infatti bastata l'aggiunta di qualche routine ROM per la gestione di questi tasti, a fare in modo che qualche programma per C64 non funzionasse bene sul C128: si è quindi preferito lasciare le cose esattamente così come sono sul C64.
Nel modo 128 l'interruttore 40/80 DISPLAY seleziona il tipo di schermo usato come default. Questo interruttore viene letto all'accensione, alla pressione di RUN/STOP+RESTORE o quando viene premuto il pulsante di RESET (vicino all'interruttore di accensione). Esso non ha alcuna funzione in modo C64, dal momento che non sono disponibili le 80 colonne, ancora per questioni di perfetta compatibilità.
Sia in modo 128 che in modo CP/M, lo stesso chip VIC usato nel C64 visualizza 40 colonne, grafica e SPRITE. Lo schermo a 40 colonne può essere visualizzato solamente su un apparecchio TV o su un monitor composito, ma non su un monitor RGB. Quest'ultimo infatti visualizza un numero doppio di pixel e caratteri rispetto al modo 40 colonne, con l'estrema purezza dei colori ottenuta separando il segnale nelle sue tre componenti: rossa, verde e blu. Un segnale composito possiede invece queste tre componenti mischiate insieme, fatto che rende difficile una netta separazione dei tre colori.
Per le 80 colonne viene usato uno speciale chip video, e lo schermo a 80 colonne può essere visualizzato soltanto su un monitor RGB. Nel modo 80 colonne sono disponibili tutti i sedici colori (benché il colore che si ottiene premendo i tasti Commodore ed 1, normalmente arancione, appaia come un violetto scuro), insieme al reverse video ed alla sottolineatura. A differenza del modo 40 colonne, sono disponibili ben 512 caratteri: ciò significa che è possibile visualizzare contemporaneamente sullo schermo caratteri maiuscoli, minuscoli e tutti i caratteri grafici ottenibili da tastiera.
Questo chip a 80 colonne è dedicato solo al testo: non consente SPRITE o grafici a mappa di bit. È possibile tuttavia ridefinire i caratteri e creare quindi una piccola finestra di 640x48 pixel, che simula una paginetta grafica.
Il chip video a 80 colonne usa ben 16 Kbyte di memoria di schermo, che tuttavia lasciano inalterata la memoria disponibile nel modo 128: quest'ultimo dispone quindi in realtà di 144 Kbyte di RAM.
Esistono tre modi per passare dalle 40 alle 80 colonne: spostare l'interruttore 40/80 e premere RUN/STOP + RESTORE, premere ESC-X in BASIC, o dare il comando SCREEN 0 per 40 colonne e SCREEN 5 per 80 colonne. Questi due schermi sono comunque indipendenti e se per ipotesi si avessero collegati al computer due diversi monitor, la stampa su uno schermo lascerebbe inalterato l'altro, e viceversa.
Il monitor Commodore 1902 è ideale per il C128, in quanto comprende un video composito a colori, un video RGB compatibile IBM ed un video RGB analogico (per futuro uso con l'"Amiga Lorraine"). Con il 1902 è possibile spostare manualmente un interruttore per poter regolare il monitor sul modo di schermo che abbiamo scelto con il computer. Ciò può sembrare alquanto laborioso, ma la Commodore ritiene – pensiamo giustamente – che si lavori per un tempo ragionevole in un modo piuttosto che in un altro, senza richiedere quindi frequentissimi passaggi dalle 40 alle 80 colonne e viceversa.
Il nuovo drive 1571 amplifica ulteriormente la potenza del C128. Nel modo C64 il 1571 si comporta esattamente come un 1541. Il 1571 che abbiamo provato non era totalmente ROM-compatibile con il 1541 (il nostro programma TURBODISK, ad esempio, non "girava"), ma abbiamo ricevuto assicurazioni che la compatibilità con il 1541 sarebbe stata in seguito migliorata.
Nel modo 128 il 1571 mostra la sua vera potenza, con capacità di immagazzinamento dati di 360 Kbyte (al contrario del 1541 che arriva a circa 170 Kbyte) e velocità di trasferimento dati da sette a dieci volte superiore.
L'aumentata capacità di immagazzinamento dati è dovuta al progetto "a doppia faccia" del 1571, con due diverse testine di lettura/scrittura, una per ogni lato del disco. Naturalmente, ciò obbliga all'uso di dischi a doppia faccia, leggermente più costosi di quelli normali.
È ancora possibile usare un 1541 in modo 128, e lo stesso 1571 può essere programmato per essere compatibile con il 1541 anche in modo 128. In questo modo non è necessario trasferire programmi precedentemente registrati con il 1541 su dischi in formato 1571/128.
Altre periferiche Commodore, come modem o stampanti, funzionano anche in modo 128, senza quindi dover richiedere la sostituzione dell'intero sistema.
Il 1571 è ottimizzato inoltre per il modo CP/M (anche qui è comunque possibile utilizzare il 1541). In modo CP/M il 1571 può immagazzinare su disco fino a 410 Kbyte di dati.
La Commodore ha sviluppato una nuova versione del CP/M, chiamata CP/M Plus, che rende possibile alle nuove applicazioni scritte in CP/M l'accesso alla grafica del chip VIC, agli SPRITE, alle 80 colonne a colori RGB ed al sintetizzatore sonoro SID: possibilità alquanto interessanti per un computer CP/M!
Al contrario del CP/M su cartuccia per C64, il CP/M Plus è una vera implementazione "alla fonte" del microprocessore Z80: le intere risorse del sistema sono quindi accessibili al CP/M stesso. La Commodore è attualmente molto impegnata nella conversione di dischi CP/M nel formato 1541, in modo che essi possono essere caricati sia dal C128 che dal CP/M su cartuccia del C64. Il nuovo 1571 può comunque essere programmato per per poter leggere molti diversi formati di disco: ad esempio, può essere usato un apposito programma di configurazione per fare in modo che il drive legga i formati più comuni del CP/M, compresi i dischi formattati dai computer Osborne o Kaypro.
Ottimo è il BASIC 7.0 del modo 128: ci pare la versione del BASIC più perfezionata fino ad ora implementata su personal computer, meglio ancora dell'Advanced BASIC IBM. Il BASIC 7.0 combina in pratica il meglio del BASIC V2, del Simon's BASIC, della Super Expander 64, del BASIC 3.5 (C16-Plus/4) e dei comandi per il drive del BASIC 4.0, comprendendo inoltre una serie di nuovi comandi studiati appositamente per il C128.
Esistono più di 80 nuovi comandi e funzioni, e gli interi 128 Kbyte sono disponibili per la programmazione: 64 Kbyte "puliti" per i programmi BASIC e 64 Kbyte per la memorizzazione di variabili, stringhe e matrici (meno la memoria usata dal sistema operativo e la mappa dello schermo a 40 colonne). L'unica particolarità, forse dimenticata in questo perfezionamento del BASIC, è la lunghezza dei nomi di variabile, ancora limitata ai primi due caratteri significativi.
Sono compresi, come detto, tutti i comandi per il drive del BASIC 4.0, tali da permettere ai possessori del C128 di eseguire alcuni dei programmi scritti per il CBM 4032/8032. Questi comandi aboliscono la necessità delle solite OPEN15, 8, 15 : PRINT # 15, "comando" : CLOSE15.
Molti dei comandi per il drive possono essere usati con sistemi a dischi doppi (i drive chiamati 0 ed 1) e con molteplici drive indirizzati con diversi numeri di periferica. SHIFT+RUN/STOP assume come default il disk drive, caricando e mandando in esecuzione il primo programma presente sul disco.
DLOAD e DSAVE sono usati per caricare e salvare programmi BASIC da/su disco, e CATALOG o DIRECTORY listano sullo schermo la directory senza cancellare il programma in memoria. SCRATCH permette di cancellare dei file su disco, domandando – prima di procedere – ARE YOU SURE?, mentre HEADER viene usato per formattare un disco nuovo.
COLLECT esegue la validate, liberando tutti i blocchi male allocati, COPY e CONCAT permettono di copiare o combinare file sul medesimo disco o tra due drive del medesimo sistema (ma non con drive contraddisti da un diverso numero di periferica). Anche BACKUP può essere usato solo con un sistema a disco doppio.
APPEND permette di aggiungere dati

ad un file esistente, DOPEN e DCLOSE rendono più facile la gestione file e RECORD rende un divertimento usare i "famigerati" file relativi.
Le variabili riservate DS e DS$ permettono di leggere il canale degli errori, e DCLEAR chiude tutti i canali rimasti aperti con il disco.
È presente inoltre una serie completa di comandi per l'aiuto nella programmazione: AUTO genera automaticamente i numeri di linea, DELETE cancella linee di programma, HELP mostra quale istruzione BASIC ha provocato un messaggio d'errore.
RENUMBER rinumera completamente un programma, compresi i GOTO e GOSUB; TRON e TROFF permettono di entrare ed uscire dal modo "trace" e KEY visualizza le definizioni correnti dei tasti funzione, permettendo di modificarle a piacere.
Le funzioni HEX$ e DEC permettono di convertire numeri decimali in esadecimali e viceversa e, oltre alle AND e OR, esiste anche l'operatore logico XOR di OR esclusivo.
Gli entusiasti della programmazione strutturata non saranno più obbligati ad usare l'odiato GOTO: la condizione IF/THEN è ora dotata anche di ELSE; BEGIN/BEND permettono di definire un gruppo di linee che verranno eseguite solo se la precedente condizione IF/THEN ha dato un risultato vero.
DO:LOOP UNTIL, DO:LOOPWHILE, DO UNTIL:LOOP e DO WHILE:LOOP eseguono una serie di linee mentre una certa condizione è vera, o fino a che una certa condizione si dimostra falsa. EXIT permette di uscire da un ciclo in qualsiasi momento.
RESTORE può ora essere fatto seguire da un numero di linea, per iniziare la lettura da una qualsiasi linea DATA. TRAP passa l'esecuzione del programma ad un numero di linea specificato, se si è verificato un errore: il programma può esaminare il tipo di errore attraverso la variabile riservata ER, il numero di linea in cui esso si è verificato attraverso la variabile riservata EL ed il messaggio d'errore attraverso la funzione ERR$. Dopo aver "gestito" l'errore RESUME ripassa il controllo all'istruzione successiva all'errore stesso, oppure ad una linea di programma specificata.
La gestione delle stringhe è semplificata dall'istruzione INSTR, che trova la posizione di una sottostringa all'interno di una stringa più lunga. PRINT USING permette di definire il formato di stampa, rendendo molto più facile incolonnare correttamente numeri e creare tabelle.
WINDOW crea una finestra con scorrimento indipendente dal resto dello schermo, e può essere usata per simulare in modo semplice una gestione delle finestre tipo Macintosh.
Più nessun bisogno di POKE per il controllo del chip sonoro SID: il BASIC 7.0 comprende molti comandi per ottenere musica ed effetti sonori.
SOUND definisce frequenza, durata e forma d'onda di un suono. PLAY è un mini-linguaggio essa stessa: può essere usata per suonare stringhe di note, specificare nomi di note, durate, toni, ecc. PLAY può essere usata per sincronizzare musica a tre voci, regolare i filtri e controllare il volume individuale di ciascuna voce.
Ogni voce può essere indirizzata da una serie di inviluppi predefiniti per simulare fino a dieci diversi strumenti musicali, che possono a loro volta essere personalizzati con ENVELOPE. Anche i filtri possono essere personalizzati con FILTER, ed è possibile regolare il volume con VOL, ed il ritmo con TEMPO.
Il BASIC 7.0 offre un ricco vocabolario di comandi indirizzati alla grafica: GRAPHIC viene usato per entrare sia in pagina grafica multicolore 160x200, sia in bitmap 320x200, sia in pagina testo a 40 colonne che in pagina testo a 80 colonne. GRAPHIC permette di definire una finestra di testo in pagina grafica, e può sia pulire lo schermo che lasciarlo inalterato. Per la pulizia dello schermo può essere usato anche SCNCLR.
Quando si entra in modo grafico l'inizio del BASIC viene spostato oltre la fine della pagina grafica. GRAPHIC CLR viene usato per disallocare la memoria usata dalla pagina grafica e RGR restituisce il numero del modo grafico corrente.
DRAW plotta un punto singolo o una linea per creare disegni complessi. LOCATE permette di definire la posizione del cursore grafico senza plottare alcun punto, BOX plotta rettangoli vuoti o pieni, con qualsiasi inclinazione. CIRCLE traccia cerchi, ellissi, archi e qualsiasi poligono, con qualsiasi angolo di rotazione. È possibile inserire del testo in qualsiasi punto della pagina grafica con l'istruzione CHAR. CHAR si può usare anche in pagina testo per simulare l'istruzione PRINT AT (mancante).
COLOR definisce il contenuto dei registri colore e RCOLOR legge il colore assegnato ai registri stessi. PAINT può riempire qualsiasi forma chiusa con un colore qualunque, SSHAPE è in grado di leggere qualsiasi blocco dello schermo ed immagazzinarlo in una stringa. Quest'ultima può poi essere ricopiata in qualsiasi posizione dello schermo con GSHAPE.
Un pixel può essere "testato" con la funzione RDOT, che restituisce il colore del pixel alla riga e colonna specificate. Il comando WIDTH definisce le dimensioni dei pixel plottati: un valore 2 raddoppia la larghezza di tutti i pixel. Infine il comando SCALE permette di trattare la pagina grafica come se avesse dimensioni di 1024x1024 pixel, ricalcolando automaticamente la scala delle figure disegnate.
Il BASIC 7.0 non sarebbe certamente completo senza comandi per gli SPRITE, e ne possiede infatti un gran numero. Per prima cosa esso comprende un semplice SPRITE editor. Basta digitare SPRDEF e sullo schermo appare una griglia. Dopo aver inserito il numero dello SPRITE che si intende definire si possono usare i tasti cursore ed i tasti numerici da 1 a 4 per definire sulla griglia i pixel accesi e spenti dello SPRITE ed i loro colori.



Il Commodore 128 è una macchina "tre in uno", contenendo un normale C64, un C64 con 128 Kbyte e un computer CP/M Z80.

Lo SPRITE viene quindi immagazzinato in un'area riservata di memoria, che può essere salvata su disco con il comando BSAVE, per essere in seguito richiamata con BLOAD. Tutto ciò elimina la necessità di lunghe serie di istruzioni DATA.
Per definire i parametri dello SPRITE si usa l'omonima istruzione SPRITE.
Questo comando abilita lo SPRITE, ne definisce colore, priorità, posizione iniziale X ed Y, e tipo di visualizzazione (normale o multicolore).
MOVSPR permette di muovere lo SPRITE sullo schermo semplicemente specificandone velocità di movimentp ed angolo del moto. Mentre lo SPRITE si muove, il programma può procedere nel compiere altre operazioni (ciò è possibile perché la posizione degli SPRITE viene aggiornata durante l'interruzione IRQ).
Mentre lo SPRITE si muove è possibile leggere la sua posizione con RSPPOS. Si può inoltre trasferire la forma dello SPRITE in qualsiasi stringa, o copiarla da una stringa in un qualsiasi SPRITE. Combinando SSHAPE con GSHAPE, è possibile leggere un qualsiasi blocco di schermo e trasformarlo in uno SPRITE, quindi duplicare il disegno dello SPRITE stesso in qualsiasi altra parte dello schermo.
SPRCOLOR definisce i registri per il formato multicolore, che vengono a loro volta letti dalla funzione RSPRCOLOR.
L'istruzione COLLISION passa il controllo ad un numero di linea specificato, se due SPRITE entrano in contatto, o se uno SPRITE tocca qualche carattere visualizzato sullo schermo. Il programma sarà in grado di sapere che cosa ha provocato la collisione, attraverso la funzione BUMP.
Opportuni comandi sono disponibili per il controllo di joystick e paddle: JOY restituisce la posizione di ciascun joystick, POT legge una delle quattro paddle collegabili e PEN legge le coordinate X e Y della penna ottica.
Infine ci sono dei comandi con varie funzioni: SLEEP si usa come ciclo di ritardo con pausa da 1 a 65353 secondi GETKEY funziona come GET,ma attende fino a che venga premuto un tasto. GO64 entra in modo C64, ma domanda prima ARE YOU SURE?, dal momento che tutto ciò che è contenuto in memoria nel modo 128 andrebbe perduto.
Il BASIC 7.0 possiede praticamente tutti i comandi di cui un programmatore potrà mai aver bisogno. Il loro numero aumenta sicuramente il tempo necessario ad imparare la programmazione del C128, ma, una volta superato questo trascurabile ostacolo, programmare diventa semplice come bere un bicchiere d'acqua.
Il C128 è il logico miglioramento del C64: senza sacrificare la compatibilità con quest'ultimo, offre tutto ciò che un appassionato può desiderare.
Il BASIC 7.0 fornisce la gran libertà di programmare senza dover ad ogni passo consultare mappe di memoria o ricorrere a routine in linguaggio macchina.
Il video a 80 colonnne, il microprocessore a 2 MHz, i 128 Kbyte di memoria (espandibili teoricamente fino a 1 Mbyte), il CP/M Plus ed il veloce drive a doppia faccia rendono il C128 un potente computer anche gestionale, competitivo con concorrenti molto più costosi, quali Apple ed IBM. Come sempre, logicamente, dovremo attendere che venga prodotto del software che tragga profitto da queste sue caratteristiche. Benché sia già possibile usare il software esistente per il C64 ed il CP/M, sembra che, almeno per un po', i programmi in BASIC 7.0 dovremo scriverceli da soli.



Digger

di **R. Long**
trad. ed adatt.
di **L. Priotto**

Nel mondo sotterraneo che è il vostro universo, le strane creature sono pressoché invincibili mentre si aprono la strada ingoiando terra. L'unica difesa è cercare di attirarle strategicamente sotto un masso pericolante. Un gioco completamente in linguaggio macchina per VIC 20 e C64.



Lo scopo del gioco è evitare tre strane creature, che vi seguono nel vostro vagabondare sottoterra. Il gioco inizia premendo F1. L'unico modo per eliminare le creature è scavare la terra sottostante una delle tre rocce, facendo in modo che queste cadano fino a toccare nuovamente il suolo, travolgendo tutte le creature che si trovano sulla loro strada.
Le creature possono inoltre essere temporaneamente immobilizzate, usando verso di loro la vostra speciale pompa ad aria. Il getto d'aria viene emesso premendo il pulsante di fuoco mentre vi state muovendo verso l'obiettivo. Una volta che la micidiale pompa è entrata in contatto con la creatura premete F7 per attivarla: dovrete attendere fino a che la creatura sia stata totalmente "gonfiata", mentre le altre continueranno a darvi la caccia.
L'uso del pulsante di fuoco richiede un minimo di esperienza: per attivare la pompa occorre essere in movimento e premere contemporaneamente il pulsante di fuoco. Una volta eliminate tutte e tre le creature inizierete un secondo schermo con la pressione della barra spaziatrice: il gioco continuerà con punteggi sempre più alti. Per ogni schermo completato compare un piccolo fiore nella parte alta dello schermo, mentre se una creatura vi tocca, il punteggio ritorna a zero e tutti i fiori vengono eliminati.

Note alla versione per C64

Nella versione per C64 Digger è un felice elfo che vive in un mondo sotterraneo, mangiando ciò che lo circonda. Egli ha un solo problema: è continuamente circondato da tre Troll che scavano tutt'attorno a lui. L'unica difesa del povero Digger è cercare di fare in modo che i Troll scavino la terra sottostante alcune grosse rocce: queste possono cadere sull'ignaro Troll, facendolo scomparire per un certo tempo. Digger deve essere comunque molto attento a non toccare nessun Troll e nessuna roccia, altrimenti tutti i Troll precedentemente scomparsi ricompariranno immediatamente. Guadagnate 10 punti per ogni Troll scomparso e 100 punti se eliminate tutti i Troll presenti sullo schermo. Con il joystick collegato in porta 2 potete muovere Digger sia orizzontalmente che verticalmente; esso continuerà a muoversi fino a che lo fermate, posizionando diagonalmente il joystick.
In qualunque momento si può fermare il gioco premendo SHIFT-LOCK. Si noti che la versione per C64 non è dotata della possibilità di distruggere i Troll tramite la pompa ad aria.
Il gioco utilizza la grafica ad alta risoluzione e gli SPRITE per visualizzare il mondo sotterraneo. Un'apposita routine di interruzione raster permette di avere contemporaneamente sullo schermo grafica in alta risoluzione e testo.

Note per il caricamento e la digitazione

Per il C64, se si dispone della cassetta allegata alla rivista, è sufficiente posizionarsi con il nastro in corrispondenza dell'inizio del programma e digitare:

LOAD"",1,1

(attenzione: non inserire spazi tra gli apici!).
Il programma andrà quindi mandato in esecuzione digitando:

SYS 49152

Ricordiamo che per conoscere il punto esatto di inizio dei vari programmi su cassetta è sufficiente inserire quest'ultima completamente riavvolta nel registratore, azzerare il contanastro e digitare:

LOAD"*"

Dal momento che sicuramente non esiste sulla cassetta alcun programma di nome "*", il computer passerà in rassegna tutti i programmi incontrati, visualizzandone il nome con il messaggio:

FOUND XXXX

Sarà quindi utile annotarsi su un foglio i nomi dei programmi e la relativa posizione del contanastro (sottrarre tre o quattro unità dal valore corrispondente all'apparizione del messaggio FOUND...).

Gli abbonati alla sola rivista, sempre con riferimento alla versione per C64, dovranno invece usare il programma MLX (vedere nelle pagine verdi) per digitare il listato in linguaggio macchina. La procedura da seguire, in questo caso, è: caricare in memoria MLX (che sarà stato precedentemente digitato e salvato) e rispondere alle sue richieste circa gli indirizzi iniziale e finale con:

indirizzo iniziale: 49152
indirizzo finale : 52187

Dopo aver digitato e salvato il programma tramite MLX si potranno seguire per il caricamento le istruzioni date più sopra per i possessori di cassetta.
Per quanto riguarda la versione per VIC 20 il programma "gira" solo sul VIC inespanso, quindi occorre disinserire qualunque eventuale espansione di memoria.
"Digger" per VIC 20, pur essendo totalmente in linguaggio macchina, si carica e si manda in esecuzione come un qualsiasi programma BASIC, con LOAD seguito da RUN.
I possessori di VIC 20 abbonati alla sola rivista dovranno usare il programma MLX (vedere nelle pagine verdi) per digitare il listato in linguaggio macchina. La procedura da seguire, in questo caso, è la seguente:
1. Spegnere il computer ed inserire un'espansione di memoria da almeno 8 Kbyte.
2. Accedere il computer e digitare in modo diretto le seguenti istruzioni:

POKE 36869,240:POKE 36866,PEEK (36866) OR128
POKE 648,30 : POKE 44,32 : POKE 32*256,0:CLR

3. Caricare in memoria MLX per VIC 20 (che sarà stato precedentemente digitato e salvato) e rispondere alle sue richieste circa gli indirizzi iniziale e finale con:

indirizzo iniziale: 4097
indirizzo finale : 7684

Dopo avere digitato e salvato il programma tramite MLX si potranno seguire per il caricamento le istruzioni date più sopra per i possessori di cassetta.

Digger versione per VIC 20

```
4097 :011,016,001,000,158,052,239
4103 :049,049,048,000,000,000,153
4109 :000,076,228,025,169,143,142
4115 :141,015,144,169,255,141,116
4121 :005,144,169,111,141,014,097
4127 :144,162,000,169,031,157,182
4133 :000,030,169,003,157,000,140
4139 :150,232,224,066,208,241,140
4145 :162,000,169,030,157,066,121
4151 :030,157,000,031,169,008,194
4157 :157,066,150,157,000,151,230
4163 :232,208,237,234,162,000,116
4169 :173,040,145,201,022,176,062
4175 :249,160,000,217,060,003,000
4181 :240,242,200,192,007,208,150
4187 :246,157,060,003,232,224,245
4193 :006,208,229,162,000,173,107
4199 :040,145,201,019,176,249,165
4205 :160,000,217,067,003,240,028
4211 :242,200,192,007,208,246,186
4217 :157,067,003,232,224,006,042
4223 :208,229,096,234,234,234,082
4229 :162,000,188,067,003,185,226
4235 :000,017,133,001,185,020,239
4241 :017,133,002,188,060,003,036
4247 :169,000,145,001,165,002,121
4253 :024,105,120,133,002,169,198
4259 :008,145,001,232,224,003,008
4265 :208,220,230,247,165,247,206
4271 :201,002,144,004,169,000,183
4277 :133,247,162,000,189,046,190
4283 :017,208,036,188,070,003,197
4289 :185,000,017,133,001,185,202
4295 :020,017,133,002,188,063,110
4301 :003,189,043,017,024,101,070
4307 :247,145,001,165,002,024,027
4313 :105,120,133,002,189,040,038
4319 :017,145,001,232,224,003,077
4325 :208,210,173,049,017,208,070
4331 :010,169,010,133,000,032,077
4337 :000,018,032,133,018,076,006
4343 :179,018,169,224,141,049,003
4349 :017,096,234,066,088,110,096
4355 :132,154,176,198,220,242,101
4361 :008,030,052,074,096,118,131
4367 :140,162,184,206,228,030,197
4373 :030,030,030,030,030,030,201
4379 :030,030,031,031,031,031,211
4385 :031,031,031,031,031,031,219
4391 :031,001,003,005,003,003,085
4397 :003,000,000,000,000,000,048
4403 :251,251,251,016,016,016,084
4409 :000,000,000,000,000,000,057
4415 :032,032,032,032,032,032,255
4421 :032,033,035,032,000,051,252
4427 :018,051,179,059,247,204,065
4433 :238,212,206,174,197,172,000
4439 :204,237,092,204,202,172,174
4445 :204,180,232,204,236,093,218
4451 :072,192,205,076,146,104,126
4457 :196,200,206,092,198,216,189
4463 :204,050,134,053,171,118,073
4469 :119,051,224,051,005,019,074
4475 :011,051,243,050,195,179,084
4481 :031,187,059,162,242,049,091
4487 :010,034,030,055,145,113,010
```

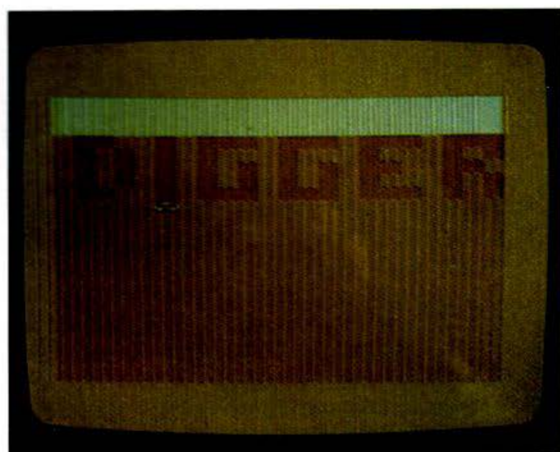
```
4493 :083,058,049,094,207,196,060
4499 :139,076,076,140,200,105,115
4505 :132,236,196,143,220,085,141
4511 :076,206,204,140,076,140,233
4517 :207,238,235,204,140,220,129
4523 :205,204,206,140,072,115,089
4529 :023,147,053,049,131,055,123
4535 :205,050,193,114,066,177,220
4541 :101,019,179,063,133,051,223
4547 :122,098,181,115,002,051,252
4553 :169,055,063,091,107,035,209
4559 :042,196,036,079,204,201,197
4565 :200,204,140,236,204,092,009
4571 :209,204,204,204,204,204,168
4577 :233,205,220,205,136,234,178
4583 :232,094,132,204,194,140,203
4589 :236,204,206,050,115,050,074
4595 :023,035,019,119,147,039,113
4601 :179,058,059,049,230,243,043
4607 :240,173,066,003,141,080,190
4613 :003,173,073,003,141,081,223
4619 :003,169,127,141,034,145,118
4625 :173,032,145,041,128,208,232
4631 :025,169,014,133,000,238,090
4637 :080,003,173,080,003,201,057
4643 :022,208,003,206,080,003,045
4649 :169,255,141,034,145,076,093
4655 :132,018,169,255,141,034,028
4661 :145,173,017,145,041,016,078
4667 :208,020,169,012,133,000,089
4673 :206,080,003,173,080,003,098
4679 :201,255,208,003,238,080,032
4685 :003,076,132,018,173,017,240
4691 :145,041,008,208,020,169,162
4697 :016,133,000,238,081,003,048
4703 :173,081,003,201,020,208,013
4709 :003,206,081,003,076,132,090
4715 :018,173,017,145,041,004,249
4721 :208,017,169,018,133,000,146
4727 :206,081,003,173,081,003,154
4733 :201,255,208,003,238,081,087
4739 :003,096,238,050,017,173,196
4745 :050,017,201,002,144,005,044
4751 :169,000,141,050,017,172,180
4757 :081,003,185,000,017,133,056
4763 :001,185,020,017,133,002,001
4769 :172,080,003,177,001,201,027
4775 :028,176,005,096,234,234,172
4781 :234,234,032,217,018,096,236
4787 :172,073,003,185,000,017,117
4793 :133,001,185,020,017,133,162
4799 :002,172,066,003,165,000,087
4805 :024,109,050,017,145,001,031
4811 :165,002,024,105,120,133,240
4817 :002,169,008,145,001,076,098
4823 :250,018,172,073,003,185,148
4829 :000,017,133,001,185,020,065
4835 :017,133,002,172,066,003,108
4841 :169,029,145,001,173,080,062
4847 :003,141,066,003,173,081,194
4853 :003,141,073,003,096,160,209
4859 :000,162,000,232,208,253,082
4865 :200,192,064,208,246,162,049
4871 :000,188,067,003,200,192,145
4877 :020,240,042,185,000,017,005
4883 :133,001,185,020,017,133,252
4889 :002,188,060,003,177,001,200
4895 :201,030,208,018,169,251,140
4901 :157,051,017,076,058,019,159
4907 :068,073,071,071,069,082,221
4913 :234,076,058,019,169,255,092
4919 :157,057,017,232,224,003,233
4925 :208,201,162,000,189,057,110
4931 :017,240,088,189,054,017,160
4937 :240,006,234,222,054,017,078
4943 :208,077,188,067,003,200,054
4949 :192,020,240,054,185,000,008
4955 :017,133,001,185,020,017,208
4961 :133,002,188,060,003,177,148
4967 :001,201,030,240,035,188,030
4973 :067,003,185,000,017,133,002
4979 :001,185,020,017,133,002,217
4985 :188,060,003,169,029,145,203
4991 :001,254,067,003,189,051,180
4997 :017,141,012,144,222,051,208
5003 :017,076,158,019,169,000,066
5009 :157,057,017,169,016,157,206
5015 :054,017,169,255,141,060,079
5021 :017,232,224,003,208,158,231
5027 :162,000,189,057,017,201,021
5033 :255,240,010,232,224,003,109
5039 :208,244,169,000,141,012,181
5045 :144,032,074,022,076,223,240
5051 :019,169,005,157,043,017,085
5057 :076,223,020,169,003,157,073
5063 :043,017,076,223,020,234,044
5069 :234,234,234,234,234,234,073
5075 :234,234,234,234,234,234,079
5081 :234,234,234,234,234,234,085
5087 :162,000,189,060,003,205,074
5093 :066,003,208,011,189,067,005
5099 :003,205,073,003,208,003,218
5105 :076,189,023,232,224,003,220
5111 :208,232,076,025,020,152,192
5117 :072,138,072,162,004,254,187
5123 :062,017,189,062,017,201,039
5129 :042,208,008,169,032,157,113
5135 :062,017,202,208,238,104,078
5141 :170,104,168,096,173,060,024
5147 :017,240,026,206,060,017,081
5153 :173,060,017,201,252,240,208
5159 :008,169,224,141,013,144,226
5165 :076,056,020,169,000,141,251
5171 :013,144,141,060,017,234,148
```

```
5177 :234,234,230,248,165,248,136
5183 :201,003,144,004,169,000,072
5189 :133,248,166,248,189,046,075
5195 :017,240,003,076,253,020,172
5201 :189,063,003,133,249,189,139
5207 :070,003,133,250,165,162,102
5213 :041,001,240,063,165,249,084
5219 :205,066,003,240,056,176,077
5225 :027,230,249,164,250,185,186
5231 :000,017,133,001,185,020,211
5237 :017,133,002,164,249,177,091
5243 :001,201,010,176,002,198,199
5249 :249,076,188,019,198,249,084
5255 :164,250,185,000,017,133,116
5261 :001,185,020,017,133,002,243
5267 :164,249,177,001,201,010,181
5273 :176,002,230,249,076,196,058
5279 :019,165,250,205,073,003,106
5285 :240,086,144,027,198,250,086
5291 :164,250,185,000,017,133,152
5297 :001,185,020,017,133,002,023
5303 :164,249,177,001,201,010,217
5309 :176,002,230,250,076,223,122
5315 :020,230,250,164,250,185,014
5321 :000,017,133,001,185,020,045
5327 :017,133,002,164,249,177,181
5333 :001,201,010,176,002,198,033
5339 :250,076,223,020,188,070,022
5345 :003,185,000,017,133,001,052
5351 :185,020,017,133,002,188,008
5357 :063,003,169,029,145,001,135
5363 :165,250,157,070,003,165,029
5369 :249,157,063,003,032,150,135
5375 :023,162,000,189,046,017,180
5381 :208,019,189,063,003,205,180
5387 :066,003,208,011,189,070,046
5393 :003,205,073,003,208,003,000
5399 :076,189,023,232,224,003,002
5405 :208,227,173,049,017,240,175
5411 :004,076,019,022,234,173,051
5417 :017,145,041,032,240,003,007
5423 :076,019,022,173,066,003,150
5429 :141,080,003,173,073,003,014
5435 :141,081,003,165,000,201,138
5441 :010,240,235,165,000,201,148
5447 :012,208,013,169,001,141,103
5453 :082,003,169,255,141,083,042
5459 :003,076,133,021,201,014,019
5465 :208,011,169,001,141,082,189
5471 :003,141,083,003,076,133,022
5477 :021,201,016,208,013,169,217
5483 :002,141,082,003,169,001,249
5489 :141,083,003,076,133,021,058
5495 :201,018,208,007,169,002,212
5501 :141,082,003,169,255,141,148
5507 :083,003,173,082,003,201,164
5513 :001,240,003,076,209,021,175
5519 :162,005,173,080,003,024,078
5525 :109,083,003,201,022,144,199
5531 :003,076,019,022,141,080,240
5537 :003,172,081,003,185,000,093
5543 :017,133,001,185,020,017,028
5549 :133,002,172,080,003,177,228
5555 :001,201,029,240,003,076,217
5561 :121,022,173,082,003,145,219
5567 :001,165,002,024,105,120,096
5573 :133,002,169,000,145,001,135
5579 :202,208,195,076,016,022,154
5585 :162,005,173,081,003,024,145
5591 :109,083,003,201,020,144,007
5597 :003,076,019,022,141,081,051
5603 :003,172,081,003,185,000,159
5609 :017,133,001,185,020,017,094
5615 :133,002,172,080,003,177,038
5621 :001,201,029,240,003,076,027
5627 :121,022,173,082,003,145,029
5633 :001,165,002,024,105,120,162
5639 :133,002,169,000,145,001,201
5645 :202,208,195,032,249,016,147
5651 :076,130,022,165,002,024,182
5657 :105,120,133,002,162,000,035
5663 :189,046,017,208,026,177,182
5669 :001,221,040,017,208,019,031
5675 :169,003,157,046,017,142,065
5681 :084,003,169,255,141,049,238
5687 :017,032,252,019,076,072,011
5693 :022,232,224,003,208,220,202
5699 :169,255,141,084,003,096,047
5705 :234,162,000,160,000,185,046
5711 :046,017,208,027,189,060,114
5717 :003,217,063,003,208,019,086
5723 :189,067,003,217,070,003,128
5729 :208,011,169,255,141,060,173
5735 :017,153,046,017,032,252,108
5741 :019,200,192,003,208,219,182
5747 :232,224,003,208,212,096,066
5753 :032,249,016,032,022,022,238
5759 :234,234,234,234,234,234,251
5765 :173,049,017,240,055,165,064
5771 :197,201,063,208,026,169,235
5777 :253,141,013,144,206,049,183
5783 :017,173,049,017,041,004,196
5789 :208,016,174,084,003,048,178
5795 :011,254,046,017,076,175,230
5801 :022,169,000,141,013,144,146
5807 :173,049,017,201,239,208,038
5813 :011,169,000,141,049,017,056
5819 :141,013,144,032,196,022,223
5825 :076,234,022,164,000,185,106
5831 :044,030,201,001,240,004,207
5837 :201,002,208,005,169,029,051
5843 :153,044,030,185,000,031,142
5849 :201,001,240,004,201,002,098
5855 :208,005,169,029,153,000,019
```

```
5861 :031,200,208,221,096,162,123
5867 :000,189,046,017,240,032,247
5873 :201,255,240,028,234,188,107
5879 :070,003,185,000,017,133,143
5885 :001,185,020,017,133,002,099
5891 :188,063,003,189,046,017,253
5897 :074,074,234,024,105,007,015
5903 :145,001,232,224,003,208,060
5909 :214,173,049,017,240,025,227
5915 :173,049,017,201,225,176,100
5921 :018,206,049,017,173,049,033
5927 :017,201,222,208,008,032,215
5933 :196,022,169,000,141,049,110
5939 :017,234,173,049,017,208,237
5945 :021,162,000,189,046,017,236
5951 :240,013,201,255,240,009,253
5957 :165,162,197,161,176,003,165
5963 :222,046,017,232,224,003,051
5969 :208,233,162,000,189,046,151
5975 :017,201,255,208,008,232,240
5981 :224,003,208,244,076,233,057
5987 :023,162,000,189,063,017,041
5993 :201,032,240,003,076,117,006
5999 :023,232,224,004,208,241,019
6005 :157,004,030,232,189,063,024
6011 :017,224,005,208,245,162,216
6017 :000,189,046,017,201,255,069
6023 :208,005,169,026,157,022,210
6029 :030,232,224,003,208,239,053
6035 :076,236,025,189,046,017,224
6041 :208,033,188,070,003,185,072
6047 :000,017,133,001,185,020,003
6053 :017,133,002,188,063,003,059
6059 :189,043,017,145,001,165,219
6065 :002,024,105,120,133,002,051
6071 :189,040,017,145,001,096,159
6077 :169,000,141,012,144,141,028
6083 :013,144,169,255,141,011,160
6089 :144,162,016,160,000,136,051
6095 :208,253,202,208,248,206,252
6101 :011,144,173,011,144,201,129
6107 :144,208,236,169,000,141,093
6113 :011,144,076,014,016,234,208
6119 :234,234,169,000,141,012,253
6125 :144,141,013,144,162,000,073
6131 :188,067,003,185,000,017,191
6137 :133,001,185,020,017,133,226
6143 :002,188,060,003,169,000,165
6149 :145,001,165,002,024,105,191
6155 :120,133,002,169,008,145,076
6161 :001,234,234,234,234,234,164
6167 :232,224,003,208,215,160,041
6173 :000,169,026,153,022,030,173
6179 :200,192,003,208,248,169,031
6185 :240,141,010,144,169,207,184
6191 :141,011,144,169,175,141,060
6197 :012,144,162,010,032,252,153
6203 :019,202,208,250,162,000,132
6209 :160,000,136,208,253,202,000
6215 :208,248,162,000,169,000,090
6221 :157,046,017,232,224,003,244
6227 :208,248,173,073,017,201,235
6233 :021,240,003,238,073,017,169
6239 :169,000,141,010,144,141,188
6245 :011,144,141,012,144,234,019
6251 :234,234,234,234,234,234,231
6257 :169,000,141,060,017,032,020
6263 :017,016,032,163,024,076,191
6269 :133,016,002,187,071,227,249
6275 :169,000,141,073,017,162,181
6281 :000,169,032,157,063,017,063
6287 :232,224,005,208,246,169,203
6293 :000,141,049,017,162,000,006
6299 :157,046,017,232,224,003,066
6305 :208,248,169,020,141,000,179
6311 :030,169,021,141,001,030,047
6317 :169,022,141,002,030,169,194
6323 :023,141,003,030,169,024,057
6329 :141,011,030,169,025,141,190
6335 :012,030,169,000,141,073,104
6341 :003,169,011,141,066,003,078
6347 :162,000,189,063,017,221,087
6353 :068,017,240,005,144,024,195
6359 :076,226,024,232,224,005,234
6365 :208,238,076,239,024,162,144
6371 :000,189,063,017,157,068,209
6377 :017,232,224,005,208,245,140
6383 :162,000,189,068,017,201,108
6389 :032,240,003,076,000,025,109
6395 :232,224,004,208,241,157,037
6401 :014,030,232,189,068,017,039
6407 :224,005,208,245,174,073,168
6413 :017,240,009,169,027,157,120
6419 :043,030,202,076,014,025,153
6425 :096,162,000,169,032,141,113
6431 :042,025,169,027,157,043,238
6437 :030,202,076,006,025,096,216
6443 :032,017,016,032,131,024,039
6449 :076,133,016,032,017,016,083
6455 :076,130,025,071,025,168,038
6461 :169,029,153,066,030,202,198
6467 :208,244,076,130,025,000,238
6473 :001,004,006,007,008,010,109
6479 :011,012,014,015,016,002,149
6485 :019,020,022,024,026,028,224
6491 :032,036,040,042,044,046,075
6497 :048,050,052,054,056,058,159
6503 :059,062,063,066,068,070,235
6509 :072,074,076,078,080,084,061
6515 :086,088,089,092,094,095,147
6521 :096,098,099,100,102,103,207
6527 :104,106,109,169,000,141,244
6533 :155,002,174,155,002,189,042
6539 :000,026,141,012,144,189,139
```

```
6545 :065,026,141,011,144,032,052
6551 :000,027,234,234,234,234,090
6557 :234,234,234,238,155,002,230
6563 :169,064,205,155,002,208,198
6569 :221,141,011,144,141,012,071
6575 :144,076,050,027,041,032,033
6581 :208,203,076,043,025,002,226
6587 :053,160,057,019,049,169,182
6593 :001,162,001,160,255,032,036
6599 :186,255,169,006,162,043,252
6605 :160,019,032,189,255,169,005
6611 :000,133,001,169,016,133,151
6617 :002,169,001,162,000,160,199
6623 :030,032,216,255,096,169,253
6629 :127,141,030,145,076,052,032
6635 :025,162,000,189,067,003,169
6641 :201,019,208,008,232,224,109
6647 :003,208,244,076,014,016,040
6653 :076,133,016,002,235,237,184
6659 :235,231,235,237,235,231,127
6665 :235,237,235,231,228,227,122
6671 :223,219,231,235,231,228,102
6677 :235,231,228,223,228,235,121
6683 :231,235,235,231,000,231,166
6689 :235,237,235,231,235,237,163
6695 :235,231,235,237,235,231,163
6701 :228,227,223,219,000,215,133
6707 :212,215,219,223,219,215,074
6713 :223,219,000,219,000,219,169
6719 :219,000,000,000,235,000,005
6725 :207,000,235,000,207,000,206
6731 :235,000,207,000,235,000,240
6737 :207,235,215,000,201,000,171
6743 :215,000,201,000,215,000,206
6749 :201,000,215,000,201,215,157
6755 :235,000,207,000,183,000,212
6761 :207,000,183,000,207,000,190
6767 :183,000,207,000,215,000,204
6773 :201,000,215,000,201,000,222
6779 :183,000,207,000,235,000,236
6785 :001,023,045,067,089,090,188
6791 :069,047,025,002,007,029,058
6797 :051,073,095,096,097,075,116
6803 :053,008,009,011,033,055,060
6809 :077,099,100,101,079,057,154
6815 :012,013,015,037,059,060,099
6821 :081,103,104,105,016,017,079
6827 :019,041,063,085,107,109,083
6833 :087,064,043,021,020,005,161
6839 :027,049,071,093,115,115,141
6845 :115,115,115,115,234,234,093
6851 :234,234,234,234,234,234,063
6857 :234,234,234,234,234,234,069
6863 :234,234,234,234,234,234,075
6869 :234,234,234,234,234,234,081
6875 :234,234,234,234,234,234,087
6881 :234,234,234,234,234,234,093
6887 :234,234,234,234,234,234,099
6893 :234,234,234,234,234,234,105
6899 :234,234,234,234,234,234,111
6905 :234,234,234,234,234,234,117
6911 :234,189,129,026,168,169,146
6917 :048,133,148,169,016,153,160
6923 :066,030,162,000,202,208,167
6929 :253,198,148,165,148,208,113
6935 :245,169,017,153,066,030,191
6941 :169,048,133,148,162,000,177
6947 :202,208,253,198,148,165,185
6953 :148,208,245,169,029,153,225
6959 :066,030,096,169,016,141,053
6965 :181,030,162,000,189,182,029
6971 :026,168,169,000,153,066,129
6977 :030,169,048,133,148,160,241
6983 :000,136,208,253,198,148,246
6989 :165,148,208,245,189,182,190
6995 :026,168,169,029,153,066,182
7001 :030,232,224,006,208,218,239
7007 :169,000,141,181,030,169,017
7013 :200,141,013,144,162,000,249
7019 :160,000,200,208,253,232,136
7025 :208,248,169,000,141,013,124
7031 :144,076,043,025,063,002,216
7037 :049,183,121,051,042,179,238
7043 :114,049,101,201,052,075,211
7049 :176,055,005,235,157,154,151
7055 :048,204,109,204,206,206,096
7061 :236,199,140,194,160,204,002
7067 :138,204,234,207,010,207,131
7073 :221,199,077,206,206,204,250
7079 :103,237,110,200,254,204,251
7085 :240,216,110,115,254,125,209
7091 :061,119,038,226,056,049,216
7097 :049,021,191,004,058,187,183
7103 :051,003,147,150,056,043,129
7109 :177,179,079,215,182,083,088
7115 :190,161,123,135,143,223,154
7121 :044,204,095,204,096,204,032
7127 :216,196,207,206,156,199,115
7133 :204,235,156,204,106,197,043
7139 :184,069,222,204,142,222,246
7145 :230,206,205,204,028,224,050
7151 :245,031,054,251,051,243,090
7157 :043,019,104,225,056,033,213
7163 :050,088,050,055,099,065,146
7169 :000,000,128,128,162,170,077
7175 :105,000,000,032,080,136,104
7181 :005,002,000,008,004,002,034
7187 :004,008,016,032,016,000,095
7193 :016,096,226,188,062,067,168
7199 :129,000,016,096,226,188,174
7205 :062,034,036,000,008,006,183
7211 :069,061,124,194,129,000,108
7217 :008,006,069,061,124,068,129
7223 :036,000,008,006,015,029,149
```

```
7229  :060,242,121,000,024,060,056
7235  :126,126,060,024,000,024,171
7241  :126,126,255,255,126,126,063
7247  :024,040,020,020,255,255,181
7253  :125,060,040,040,020,020,134
7259  :255,255,125,060,040,010,068
7265  :006,006,015,015,013,003,155
7271  :010,010,006,006,015,063,213
7277  :055,163,010,160,144,144,017
7283  :240,240,112,192,160,160,195
7289  :144,144,240,252,220,202,043
7295  :160,040,020,020,255,255,109
7301  :125,056,032,040,020,020,170
7307  :255,255,125,044,008,040,098
7313  :040,040,255,255,125,044,136
7319  :008,040,040,040,255,255,021
7325  :125,056,032,255,255,248,104
7331  :247,249,254,241,255,255,128
7337  :255,198,189,189,189,198,107
7343  :255,255,255,099,173,163,095
7349  :171,109,255,255,255,031,233
7355  :127,063,127,031,255,255,021
7361  :255,107,106,010,106,107,116
7367  :255,255,255,022,246,144,096
7373  :214,022,255,255,247,249,167
7379  :240,226,195,013,134,243,238
7385  :224,204,224,211,191,127,118
7391  :127,040,040,040,040,040,038
7397  :040,040,040,000,000,000,093
7403  :000,000,000,000,000,068,047
7409  :017,068,017,068,017,068,240
7415  :017,255,255,255,255,255,003
7421  :255,255,255,195,189,185,051
7427  :165,157,189,195,255,247,187
7433  :231,215,247,247,247,193,109
7439  :255,195,189,253,243,207,077
7445  :191,129,255,195,189,253,209
7451  :227,253,189,195,255,251,117
7457  :243,235,219,129,251,251,081
7463  :255,129,191,135,251,253,229
7469  :187,199,255,227,223,191,047
7475  :131,189,189,195,255,129,115
7481  :189,251,247,239,239,239,181
7487  :255,195,189,189,195,189,251
7493  :189,195,255,195,189,189,001
7499  :193,253,251,199,255,255,201
7505  :000,000,003,076,000,000,160
7511  :000,011,015,001,015,000,129
7517  :000,000,000,000,143,000,236
7523  :000,096,000,000,000,000,195
7529  :141,128,000,000,000,000,118
7535  :000,000,012,195,230,122,158
7541  :208,002,230,123,173,008,093
7547  :002,201,058,176,010,201,003
7553  :032,240,239,056,233,048,209
7559  :056,233,208,096,128,079,167
7565  :199,082,088,000,255,000,253
7571  :000,000,000,000,016,000,163
7577  :000,003,000,000,128,000,028
7583  :000,000,105,135,000,000,143
7589  :000,000,000,000,000,000,165
7595  :000,000,000,000,000,000,171
7601  :000,060,003,000,000,000,240
7607  :000,000,000,000,000,000,183
7613  :000,000,000,000,124,108,165
7619  :000,029,015,000,000,007,246
7625  :007,006,015,001,002,032,008
7631  :000,021,154,030,007,000,163
7637  :021,007,048,000,158,158,093
7643  :158,158,158,158,158,158,143
7649  :158,158,158,158,159,159,151
7655  :159,159,159,159,159,159,161
7661  :159,159,159,159,255,000,104
7667  :154,150,094,236,000,000,109
7673  :014,014,000,000,000,029,050
7679  :032,013,013,013,013,255,082
```



Digger versione per C64

```
49152  :076,080,194,169,127,141,019
49158  :013,220,173,017,208,041,166
49164  :127,141,017,208,169,035,197
49170  :141,020,003,169,192,141,172
49176  :021,003,173,026,208,009,208
49182  :001,141,026,208,096,169,159
49188  :001,141,025,208,162,093,154
49194  :160,000,173,018,208,201,034
49200  :093,144,004,162,001,160,100
49206  :032,140,041,204,142,018,119
49212  :208,173,041,204,240,022,180
49218  :162,061,202,208,253,013,197
49224  :017,208,141,017,208,169,064
49230  :008,013,024,208,141,024,240
```

```
49236 :208,076,101,192,173,017,083
49242 :208,041,223,141,017,208,160
49248 :169,021,141,024,208,169,060
49254 :001,141,026,208,173,013,152
49260 :220,041,001,240,003,076,177
49266 :049,234,076,188,254,169,060
49272 :000,141,131,205,173,030,032
49278 :208,141,119,205,041,001,073
49284 :240,005,169,001,141,131,051
49290 :205,162,000,189,009,208,143
49296 :201,235,208,011,232,232,239
49302 :224,007,144,243,169,001,170
49308 :141,131,205,173,021,208,011
49314 :041,014,208,005,169,002,089
49320 :141,131,205,174,130,205,130
49326 :189,008,208,157,067,205,240
49332 :173,016,208,061,196,192,002
49338 :240,013,169,001,157,068,066
49344 :205,076,206,192,016,000,119
49350 :032,000,064,169,000,157,108
49356 :068,205,056,189,067,205,226
49362 :233,016,157,067,205,189,053
49368 :068,205,233,000,157,068,179
49374 :205,056,189,009,208,233,098
49380 :035,157,083,205,173,217,074
49386 :203,072,173,218,203,072,151
49392 :173,233,203,072,173,189,003
49398 :204,072,173,249,203,072,195
49404 :165,253,072,165,254,072,209
49410 :189,083,205,141,233,203,032
49416 :189,067,205,141,217,203,006
49422 :189,068,205,141,218,203,014
49428 :138,072,032,167,199,104,220
49434 :170,172,189,204,177,253,167
49440 :208,031,200,177,253,208,085
49446 :026,173,008,208,201,235,121
49452 :176,019,222,109,205,208,215
49458 :019,169,001,157,109,205,198
49464 :254,009,208,032,118,193,102
49470 :076,070,193,169,050,157,009
49476 :109,205,104,133,254,104,209
49482 :133,253,104,141,249,203,133
49488 :104,141,189,204,104,141,195
49494 :233,203,104,141,218,203,164
49500 :104,141,217,203,238,130,101
49506 :205,238,130,205,173,130,155
49512 :205,201,007,176,003,076,004
49518 :171,192,169,000,141,130,145
49524 :205,096,173,021,208,141,192
49530 :171,204,173,119,205,041,011
49536 :112,240,065,169,000,141,087
49542 :155,204,141,136,205,160,111
49548 :001,189,196,192,141,064,155
49554 :205,185,249,197,013,064,035
49560 :205,045,171,204,141,021,171
49566 :208,032,239,193,173,030,009
49572 :208,041,014,072,013,155,155
49578 :204,141,155,204,104,240,194
49584 :003,238,136,205,200,192,126
49590 :004,144,212,173,155,204,050
49596 :073,255,045,171,204,141,053
49602 :021,208,173,119,205,041,193
49608 :001,141,131,205,173,171,254
49614 :204,205,021,208,240,026,086
49620 :174,136,205,024,173,134,034
49626 :205,105,010,141,134,205,250
49632 :173,135,205,105,000,141,215
49638 :135,205,202,208,236,032,224
49644 :187,194,096,173,018,208,088
49650 :208,251,173,017,208,016,091
49656 :246,096,169,128,133,251,247
49662 :169,039,133,252,169,204,196
49668 :141,128,039,160,000,177,137
49674 :251,073,255,200,145,251,161
49680 :056,165,251,233,063,141,157
49686 :041,204,165,252,233,063,212
49692 :013,041,204,240,009,230,253
49698 :251,208,226,230,252,076,253
49704 :007,194,096,162,000,160,147
49710 :016,024,032,240,255,160,005
49716 :000,185,074,194,032,210,235
49722 :255,200,192,006,208,245,140
49728 :165,162,024,105,100,197,049
49734 :162,208,252,096,068,073,161
49740 :071,071,069,082,169,147,173
49746 :032,210,255,169,000,141,121
49752 :139,205,141,140,205,032,182
49758 :003,192,169,000,141,057,144
49764 :204,141,134,205,141,135,036
49770 :205,032,250,193,032,113,163
49776 :196,032,091,195,173,030,061
49782 :208,032,043,194,032,119,234
49788 :192,032,136,197,032,069,014
49794 :196,173,131,205,201,002,014
49800 :240,007,201,001,240,210,011
49806 :076,176,194,238,057,204,063
49812 :024,169,100,109,134,205,121
49818 :141,134,205,173,135,205,123
49824 :105,000,141,135,205,032,010
49830 :187,194,169,000,141,131,220
49836 :205,076,107,194,173,141,044
49842 :002,208,251,076,184,194,069
49848 :076,122,194,169,000,141,118
49854 :132,205,173,021,208,074,235
49860 :041,007,141,133,205,160,115
49866 :002,024,078,133,205,169,045
49872 :000,109,132,205,141,132,159
49878 :205,136,016,241,160,030,234
49884 :162,002,024,032,240,255,167
49890 :169,000,174,132,205,208,090
49896 :000,032,205,189,056,173,119
49902 :134,205,237,139,205,141,019
49908 :064,205,173,135,205,237,239
49914 :140,205,013,064,205,144,253
```

```
49920 :012,173,134,205,141,139,036
49926 :205,173,135,205,141,140,237
49932 :205,160,006,162,004,024,061
49938 :032,240,255,173,140,205,039
49944 :174,139,205,032,205,189,200
49950 :160,006,162,002,024,032,160
49956 :240,255,173,135,205,174,194
49962 :134,205,032,205,189,160,199
49968 :030,162,004,024,032,240,028
49974 :255,169,000,174,057,204,145
49980 :032,205,189,096,083,067,220
49986 :079,082,069,058,084,082,008
49992 :079,076,076,083,083,067,024
49998 :082,069,069,078,083,072,019
50004 :073,045,083,067,079,082,001
50010 :069,160,024,169,000,153,153
50016 :000,212,136,016,250,169,111
50022 :015,141,024,212,169,017,168
50028 :141,005,212,141,012,212,063
50034 :169,241,141,006,212,141,000
50040 :013,212,160,000,140,137,014
50046 :205,162,000,165,162,024,076
50052 :105,002,197,162,208,252,034
50058 :238,137,205,185,222,195,040
50064 :205,137,205,208,020,200,095
50070 :185,222,195,141,000,212,081
50076 :200,185,222,195,141,001,076
50082 :212,200,169,033,141,004,153
50088 :212,189,032,196,205,137,115
50094 :205,208,020,232,189,032,036
50100 :196,141,007,212,232,189,133
50106 :032,196,141,008,212,232,239
50112 :169,033,141,011,212,165,155
50118 :162,024,105,006,197,162,086
50124 :208,252,169,032,141,004,242
50130 :212,141,011,212,173,137,072
50136 :205,201,030,144,164,096,032
50142 :002,135,033,003,135,033,051
50148 :004,135,033,005,135,033,061
50154 :006,030,025,008,030,025,102
50160 :010,049,028,011,030,025,137
50166 :012,096,022,013,031,021,185
50172 :015,096,022,017,030,025,201
50178 :018,135,033,019,135,033,119
50184 :020,135,033,021,135,033,129
50190 :022,030,025,024,030,025,170
50196 :026,049,028,027,030,025,205
50202 :028,096,022,029,031,021,253
50208 :001,097,008,010,071,006,225
50214 :011,012,007,012,233,007,064
50220 :013,097,008,015,233,007,161
50226 :017,012,007,018,071,006,181
50232 :026,071,006,027,012,007,205
50238 :028,233,007,029,097,008,208
50244 :255,173,063,205,201,005,202
50250 :144,033,160,002,185,249,079
50256 :007,073,001,153,249,007,058
50262 :136,016,245,173,248,007,143
50268 :201,135,144,005,169,128,106
50274 :141,248,007,238,248,007,219
50280 :169,000,141,063,205,238,152
50286 :063,205,096,032,102,197,037
50292 :169,147,032,210,255,160,065
50298 :005,162,001,024,032,240,074
50304 :255,160,000,185,064,195,219
50310 :032,210,255,200,192,005,004
50316 :208,245,160,028,162,001,176
50322 :024,032,240,255,160,000,089
50328 :185,070,195,032,210,255,075
50334 :200,192,006,208,245,160,145
50340 :028,162,003,024,032,240,141
50346 :255,160,000,185,076,195,017
50352 :032,210,255,200,192,007,048
50358 :208,245,160,003,162,003,195
50364 :024,032,240,255,160,000,131
50370 :185,083,195,032,210,255,130
50376 :200,192,009,208,245,160,190
50382 :013,185,088,197,153,000,074
50388 :208,170,152,041,001,208,224
50394 :007,138,153,058,204,076,086
50400 :231,196,169,000,153,058,007
50406 :204,136,016,229,169,128,088
50412 :141,248,007,169,136,141,054
50418 :249,007,141,250,007,141,013
50424 :251,007,169,138,141,252,182
50430 :007,141,253,007,141,254,033
50436 :007,160,004,185,084,197,129
50442 :153,039,208,136,016,247,041
50448 :160,003,169,007,153,043,039
50454 :208,136,016,248,169,001,032
50460 :141,037,208,169,002,141,214
50466 :038,208,169,127,141,028,233
50472 :208,141,021,208,160,015,025
50478 :169,000,153,173,204,153,130
50484 :075,204,169,015,153,031,187
50490 :205,136,016,240,169,000,056
50496 :141,065,205,141,016,208,072
```



```
50502 :169,003,141,074,204,169,062
50508 :009,141,032,208,032,187,173
50514 :194,096,001,014,005,007,143
50520 :173,173,255,130,239,120,154
50526 :100,190,060,131,255,140,202
50532 :140,140,160,000,185,025,238
50538 :201,153,000,032,200,208,132
50544 :247,160,000,185,025,202,163
50550 :153,000,033,200,208,247,191
50556 :160,000,185,025,203,153,082
50562 :000,034,200,208,247,096,147
50568 :174,065,205,189,245,197,187
50574 :141,066,205,189,249,197,165
50580 :141,064,003,173,000,220,237
50586 :072,173,064,003,201,002,157
50592 :144,031,104,032,002,199,160
50598 :152,032,231,197,032,064,106
50604 :199,152,032,231,197,238,197
50610 :065,205,173,065,205,201,068
50616 :004,144,005,169,000,141,135
50622 :065,205,096,032,241,199,004
50628 :238,065,205,104,041,015,096
50634 :073,015,240,006,141,074,239
50640 :204,076,215,197,173,074,123
50646 :204,072,032,231,197,032,214
50652 :241,199,104,072,032,231,075
50658 :197,032,241,199,104,010,241
50664 :168,185,254,197,072,185,013
50670 :253,197,072,174,066,205,181
50676 :096,000,002,004,006,001,097
50682 :002,004,008,221,198,020,191
50688 :198,031,198,221,198,127,205
50694 :198,221,198,221,198,221,239
50700 :198,042,198,221,198,221,066
50706 :198,221,198,169,099,221,100
50712 :001,208,176,003,222,001,123
50718 :208,096,169,232,221,001,189
50724 :208,144,003,254,001,208,086
50730 :096,056,189,058,204,233,110
50736 :064,141,091,204,189,059,028
50742 :204,233,001,013,091,204,032
50748 :144,013,169,064,157,058,153
50754 :204,169,001,157,059,204,092
50760 :076,083,198,254,058,204,177
50766 :208,003,254,059,204,056,094
50772 :189,058,204,233,000,141,141
50778 :091,204,189,059,204,233,046
50784 :001,013,091,204,144,012,049
50790 :173,064,003,013,016,208,067
50796 :141,016,208,076,215,198,194
50802 :173,064,003,073,255,045,215
50808 :016,208,141,016,208,076,017
50814 :215,198,056,189,058,204,022
50820 :233,025,141,091,204,189,247
50826 :059,204,233,000,013,091,226
50832 :204,176,013,169,024,157,119
50838 :058,204,169,000,157,059,029
50844 :204,076,173,198,222,058,063
50850 :204,189,058,204,201,255,249
50856 :208,003,222,059,204,056,152
50862 :189,058,204,233,000,141,231
50868 :091,204,189,059,204,233,136
50874 :001,013,091,204,144,012,139
50880 :173,064,003,013,016,208,157
50886 :141,016,208,076,215,198,028
50892 :173,064,003,073,255,045,049
50898 :016,208,141,016,208,189,220
50904 :058,204,157,000,208,096,171
50910 :076,215,198,174,066,205,132
50916 :222,031,205,240,008,169,079
50922 :000,141,030,205,076,001,175
50928 :199,173,004,220,041,031,140
50934 :056,105,001,157,031,205,033
50940 :169,001,141,030,205,096,126
50946 :032,225,198,173,030,205,097
50952 :240,046,056,189,058,204,033
50958 :237,058,204,157,107,204,213
50964 :189,059,204,237,059,204,204
50970 :157,108,204,029,107,204,067
50976 :240,012,176,005,160,008,121
50982 :076,048,199,160,004,076,089
50988 :048,199,160,000,152,157,248
50994 :075,204,032,114,199,096,002
51000 :189,075,204,168,032,114,070
51006 :199,096,032,225,198,173,217
51012 :030,205,240,034,056,189,054
51018 :001,208,237,001,208,157,118
51024 :123,204,240,012,176,005,072
51030 :160,002,076,098,199,160,013
51036 :001,076,098,199,160,000,114
51042 :152,157,173,204,032,114,162
51048 :199,096,189,173,204,168,109
51054 :032,114,199,096,056,189,028
51060 :107,204,253,123,204,141,124
51066 :064,205,189,108,204,233,101
51072 :000,013,064,205,176,007,081
51078 :189,173,204,168,076,145,065
51084 :199,189,075,204,168,152,103
51090 :072,174,065,205,173,021,088
51096 :208,061,249,197,240,003,086
51102 :032,241,199,174,066,205,051
51108 :104,168,096,173,233,203,117
51114 :041,007,072,173,233,203,131
51120 :074,074,074,168,185,229,212
51126 :200,133,253,185,255,200,128
51132 :133,254,024,104,101,253,033
51138 :133,253,165,254,105,000,080
51144 :133,254,173,217,203,041,197
51150 :007,073,007,168,185,233,111
51156 :199,141,249,203,024,173,177
51162 :218,203,101,254,133,254,101
51168 :173,217,203,041,248,141,223
51174 :189,204,096,001,002,004,214
51180 :008,016,032,064,128,174,146
```

```
51186 :066,205,056,189,058,204,252
51192 :233,024,141,217,203,189,231
51198 :059,204,233,000,141,218,085
51204 :203,056,189,001,208,233,126
51210 :050,141,233,203,141,172,182
51216 :204,169,000,170,168,024,239
51222 :185,087,200,109,172,204,211
51228 :141,233,203,200,185,087,053
51234 :200,141,237,204,162,000,210
51240 :032,075,200,238,233,203,253
51246 :232,236,237,204,208,244,127
51252 :200,185,087,200,016,004,232
51258 :200,076,021,200,238,217,242
51264 :203,208,003,238,218,203,113
51270 :200,192,124,144,202,138,046
51276 :072,152,072,032,213,200,049
51282 :104,168,104,170,096,007,219
51288 :004,000,007,001,255,010,109
51294 :001,000,005,002,255,011,112
51300 :002,000,005,001,255,012,119
51306 :001,000,005,001,255,012,124
51312 :001,000,004,001,255,013,130
51318 :001,000,004,001,255,013,136
51324 :001,000,003,001,255,014,142
51330 :001,000,003,001,255,014,148
51336 :001,000,002,001,255,015,154
51342 :001,000,002,001,255,015,160
51348 :001,000,002,001,255,015,166
51354 :001,000,002,001,255,015,172
51360 :001,000,002,001,255,015,178
51366 :001,000,003,001,255,014,184
51372 :001,000,003,001,255,014,190
51378 :001,000,004,001,255,013,196
51384 :001,000,004,001,255,013,202
51390 :001,000,005,001,255,012,208
51396 :001,000,005,002,255,011,214
51402 :002,000,007,001,000,010,222
51408 :001,000,007,004,000,032,252
51414 :167,199,172,189,204,173,038
51420 :249,203,073,255,049,253,022
51426 :145,253,096,000,064,128,144
51432 :192,000,064,128,192,000,040
51438 :064,128,192,000,064,128,046
51444 :192,000,064,128,192,000,052
51450 :064,128,192,000,064,002,188
51456 :033,034,035,037,038,039,216
51462 :040,042,043,044,045,047,011
51468 :048,049,050,052,053,054,062
51474 :055,057,058,059,060,062,113
51480 :063,000,000,000,000,000,087
51486 :000,000,252,000,003,087,116
51492 :000,013,085,192,052,016,138
51498 :112,053,085,112,213,017,122
51504 :092,213,085,092,209,085,056
51510 :028,052,000,112,053,001,044
51516 :112,013,085,192,003,087,040
51522 :000,000,252,000,000,000,062
51528 :000,000,000,000,000,000,072
51534 :000,000,000,000,000,000,078
51540 :000,000,000,000,000,000,084
51546 :000,000,000,000,000,000,090
51552 :252,000,003,087,000,013,195
51558 :005,192,053,085,112,053,090
51564 :020,112,209,085,028,209,003
51570 :081,092,208,085,092,052,212
51576 :021,092,053,001,112,013,156
51582 :085,192,003,087,000,000,237
51588 :252,000,000,000,000,000,128
51594 :000,000,000,000,000,000,138
51600 :000,000,000,000,000,000,144
51606 :000,000,000,000,000,000,150
51612 :000,000,000,000,252,000,152
51618 :003,087,000,013,085,192,030
51624 :053,021,112,052,084,112,090
51630 :208,068,092,208,085,092,159
51636 :208,068,092,052,084,112,028
51642 :053,021,112,013,085,192,150
51648 :003,087,000,000,252,000,022
51654 :000,000,000,000,000,000,198
51660 :000,000,000,000,000,000,204
51666 :000,000,000,000,000,000,210
51672 :000,000,000,000,000,000,216
51678 :000,000,252,000,003,087,052
51684 :000,013,085,192,053,001,060
51690 :112,052,021,092,208,085,036
51696 :092,209,081,092,209,085,240
51702 :028,053,020,112,053,085,085
51708 :112,013,005,192,003,087,152
51714 :000,000,252,000,000,000,254
51720 :000,000,000,000,000,000,008
51726 :000,000,000,000,000,000,014
51732 :000,000,000,000,000,000,020
51738 :000,000,000,000,000,000,026
51744 :252,000,003,087,000,013,131
51750 :085,192,053,001,112,052,021
51756 :000,112,209,085,028,213,179
51762 :085,092,213,017,092,053,090
51768 :085,112,052,016,112,013,190
51774 :085,192,003,087,000,000,173
51780 :252,000,000,000,000,000,064
51786 :000,000,000,000,000,000,074
51792 :000,000,000,000,000,000,080
51798 :000,000,000,000,000,000,086
51804 :000,000,000,000,252,000,088
51810 :003,087,000,013,085,192,222
51816 :053,001,112,213,080,112,163
51822 :213,084,028,213,021,028,185
51828 :209,085,028,052,081,112,171
51834 :053,085,112,013,065,192,130
51840 :003,087,000,000,252,000,214
51846 :000,000,000,000,000,000,134
51852 :000,000,000,000,000,000,140
51858 :000,000,000,000,000,000,146
51864 :000,000,000,000,000,000,152
```

```
51870 :000,000,252,000,003,087,244
51876 :000,013,085,192,053,081,076
51882 :112,052,084,112,212,068,042
51888 :028,213,084,028,212,068,041
51894 :028,052,084,112,053,081,080
51900 :112,013,085,192,003,087,168
51906 :000,000,252,000,000,000,190
51912 :000,000,000,000,000,000,200
51918 :000,000,000,000,000,000,206
51924 :000,000,000,000,000,000,212
51930 :000,000,000,000,000,000,218
51936 :252,000,003,087,000,013,067
51942 :065,192,053,085,112,052,021
51948 :081,112,209,085,028,213,196
51954 :021,028,213,084,028,213,061
51960 :080,112,053,001,112,013,107
51966 :085,192,003,087,000,000,109
51972 :252,000,000,000,000,000,000
51978 :000,000,000,000,000,000,010
51984 :000,000,000,000,000,000,016
51990 :000,000,000,000,000,000,022
51996 :000,084,000,001,169,000,026
52002 :006,170,064,006,238,064,070
52008 :026,170,144,026,170,144,208
52014 :026,170,144,026,254,144,042
52020 :027,255,144,006,170,064,206
52026 :006,170,064,001,169,000,212
52032 :001,084,000,001,004,000,154
52038 :001,005,000,001,064,000,141
52044 :000,000,000,000,000,000,076
52050 :000,000,000,000,000,000,082
52056 :255,000,000,000,000,084,171
52062 :000,001,169,000,006,170,184
52068 :064,006,238,064,026,170,156
52074 :144,026,170,144,026,170,018
52080 :144,026,254,144,027,171,110
52086 :144,006,254,064,006,170,250
52092 :064,001,169,000,001,084,187
52098 :000,001,004,000,001,068,204
52104 :000,000,005,000,000,000,141
52110 :000,000,000,000,000,000,142
52116 :000,000,000,000,255,000,147
52122 :000,000,000,000,000,000,154
52128 :000,000,000,000,000,000,160
52134 :000,000,003,252,000,003,168
52140 :172,000,014,175,192,254,211
52146 :178,240,239,170,176,234,135
52152 :186,175,250,175,191,063,200
52158 :170,172,000,255,240,000,003
52164 :000,000,000,000,000,000,196
52170 :000,000,000,000,000,000,202
52176 :000,000,000,000,000,000,208
52182 :000,000,255,013,013,255,238
```

In edicola il primo numero di

7 Note Bit

Oggi la musica si impara così.

SPEDIZIONE IN ABBONAMENTO POSTALE GRUPPO II/70
L. 10.000
NOTE 7 BIT
IMPARARE LA MUSICA COL C64

C64

LA MUSICA E... SUBITO ALLA TASTIERA CON IL C64

È in edicola a L. 10.000
Questa favolosa tastiera compresa nel prezzo

per C64

GRUPPO ETHOS

Il 1985 è l'anno mondiale della musica... preparati con 7 note Bit, la nuova, fantastica opera che in soli 15 fascicoli quattordicinali (ciascuno corredato da una cassetta software per Commodore 64) ti insegna veramente la musica. E puoi suonare subito

Non perdere il primo numero. Pensa, compresa nel prezzo c'è anche la stupenda tastiera professionale per il tuo Commodore 64.



GRUPPO EDITORIALE JACKSON

Milano - Londra - Madrid - San Francisco

IN COLLABORAZIONE CON SIEL

Tank per C16 e Plus/4

di **Angelo Motta**

Originariamente scritto per lo Spectrum ("Campo di Battaglia" apparso su SUPERSINC n. 2/84), il programma vi pone al comando di un carro armato con l'arduo compito di distruggere le postazioni nemiche.
Per concludere la vostra missione, però, dopo che tutte le postazioni sono state annientate dovrete far rientro alla vostra base, dove vi sarà assegnata una nuova missione, ancora più difficile.
Gli ostacoli che dovrete evitare sono mine, trappole anticarro e crateri, questi ultimi di due tipi.



Il gioco inizia con missioni facili: poche postazioni e pochi ostacoli, ma, man mano che verranno compiute le missioni il livello di difficoltà aumenterà (maggiori ostacoli e postazioni) sino a diventare quasi impossibile muoversi sul video.
Le istruzioni sono contenute all'interno del programma. Vi sono inoltre una buona quantità di REM che rendono chiara la lettura dei vari blocchi di cui è composto.
Per quanto riguarda i comandi (A=destra, 4=sinistra, 3=alto e W=basso), i tasti utilizzati possono rivelarsi scomodi al giocatore, ma sono stati intenzionalmente scelti, in quanto permettono di utilizzare il joystick nella porta 2 senza dover usare la funzione JOY, con conseguente maggior velocità al programma e quindi al movimento del carro.
Punteggio:
– 10 punti per ogni postazione distrutta;
– 100 punti per ogni missione conclusa;
– BONUS di 300 punti e l'aggiunta di un carro ad ogni cinque missioni concluse.

Tank per C16 e Plus/4

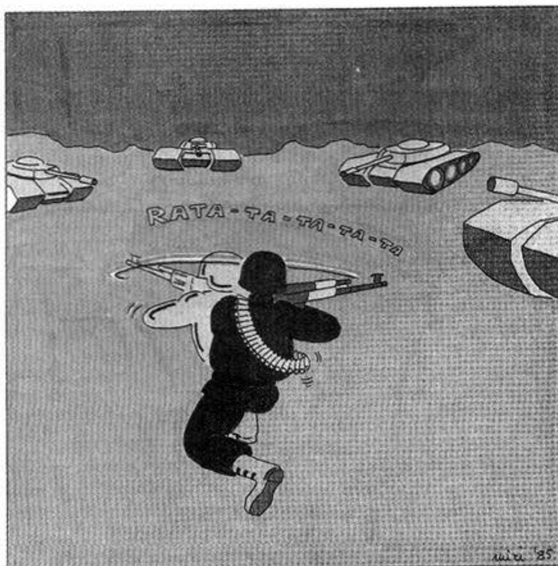
```
1000 REM TRASFERIMENTO CARATTERI IN RAM
                               :rem 158
1010 REM E MODIFICA            :rem  39
1020 :                         :rem 253
1030 POKE56,60:POKE52,60:POKE55,0:POKE51,
     0                         :rem 172
1040 T=0:FORI=1000TO1030:READA :rem 236
1050 POKEI,A:T=T+A:NEXT        :rem 175
1060 IFT=3288THEN1080          :rem 180
1070 PRINT"{CLR}ERRORE NEI DATA!!!":END
                               :rem 101
1080 SYS1000:POKE65298,0:POKE65299,60
                               :rem 102
1090 FORI=16136TO16255:READA:POKEI,A:NEXT
                               :rem  13
1190 :                         :rem   5
1200 REM DATA ROUTINE L/M PER  :rem 150
1210 REM TRASFERIMENTO SET CARATTERI
                               :rem  22
1220 :                         :rem 255
1230 DATA160,0,185,0,208,153,0,60
                               :rem 114
1240 DATA185,0,209,153,0,61,185,0
                               :rem 124
1250 DATA210,153,0,62,185,0,211,153
                               :rem 213
```

```
1260 DATA0,63,24,136,208,228,96 :rem  41
1290 :                          :rem   6
1300 REM DATA CODICI NUOVI CARATTERI
                                :rem 157
1310 REM DA COD 97 A COD 111    :rem  30
1320 :                          :rem   0
1330 DATA16,146,186,254,254,254,186,130
                                :rem 186
1340 DATA130,186,254,254,254,186,146,16
                                :rem 187
1350 DATA254,56,124,127,124,56,254,0
                                :rem  27
1360 DATA127,28,62,254,62,28,127,0
                                :rem 187
1370 DATA0,0,0,16,124,254,254,0 :rem  13
1380 DATA112,76,66,76,112,64,64,64
                                :rem 198
1390 DATA0,16,56,124,124,254,254,254
                                :rem  28
1400 DATA0,6,15,31,63,31,31,15  :rem 218
1410 DATA48,124,254,255,254,254,254,252
                                :rem 184
1420 DATA63,63,63,31,31,7,3,0   :rem 178
1430 DATA255,255,254,254,252,248,176,32
                                :rem 190
1440 DATA16,124,255,255,255,126,28,16
                                :rem  83
1450 DATA14,62,126,14,2,2,2,2   :rem 171
1460 DATA0,126,66,90,90,166,126,0
                                :rem 134
1470 DATA255,129,189,165,165,189,129,255
                                :rem   6
1490 :                          :rem   8
1500 REM PRESENTAZIONE GIOCO    :rem   2

1510 :                          :rem   1
1520 COLOR0,6,2:COLOR4,6,2      :rem 228
1530 PRINT"{CLR}":CHAR,16,3,"{YEL}T A N K
     ":CHAR,16,4,"{ 7 E}{WHT}"  :rem 214
1540 CHAR,7,8,"VUOI LE ISTRUZIONI? (S/N)"
                                :rem 115
1550 GETZ$:IFZ$="N"THEN2000     :rem  48
1560 IFZ$<>"S"THEN1550          :rem 228
1570 PRINT:PRINT"{ 2 SU}SEI AL COMANDO DI
      UN CARRO ARMATO COL"      :rem  63
1580 PRINT"COMPITO DI DISTRUGGERE TUTTE L
     E POSTA-"                  :rem 103
1590 PRINT"ZIONI NEMICHE."      :rem  80
1600 PRINT"PER MUOVERE USA:"    :rem 197
1610 PRINT"<A> PER ANDARE A SINISTRA":PRI
     NT"<4> PER ANDARE A DESTRA":rem 227
1620 PRINT"<3> PER SALIRE":PRINT"<W> PER
     SCENDERE"                  :rem 250
1630 PRINT"OPPURE USA IL JOYSTICK NELLA P
     RESA N.2"                  :rem 249
1640 PRINT"PER DISTRUGGERE LE POSTAZIONI
     PASSA SO-":PRINT"PRA DI ESSE."
                                :rem 137
1650 PRINT"FAI ATTENZIONE ALLE MINE (
     [<G>]), ALLE"              :rem 245
1660 PRINT"TRAPPOLE ANTICARRO ([<M>]) ED
     AI CRATERI."               :rem 235
1670 PRINT"QUANDO AVRAI DISTRUTTO TUTTE L
     E POSTA-"                  :rem 135
1680 PRINT"ZIONI NEMICHE DOVRAI TORNARE A
     LLA TUA"                   :rem   6
1690 PRINT"BASE ([<Z>]) PRIMA DI INIZIARE
      UN'ALTRA":PRINT"MISSIONE.":rem 249
1700 PRINT"{GIU'}PREMI <RETURN> PER INIZI
     ARE"                       :rem 195
1710 GETZ$:IFZ$<>CHR$(13)THEN1710
                                :rem 150
1900 :                          :rem  13
2000 REM INIZIALIZZAZIONE GIOCO :rem 234
2010 :                          :rem 253
2020 TK=5:SC=0:SK=0:V=3072      :rem 240
2030 COLOR0,16,7:COLOR4,16,4:VOL8
                                :rem 173
2090 :                          :rem   5
2100 REM INIZIALIZZAIONE FASE   :rem  63
2110 :                          :rem 254
2120 SK=SK+1:PRINT"{CLR}"       :rem  56
2130 FORI=3912TO3951:POKEI,68:POKEI-1024,
     0:NEXT                     :rem  62
2140 FORI=1TOSK:GOSUB9000       :rem  41
2150 POKEB,101:POKEB-1024,22    :rem  38
2160 GOSUB9000                  :rem  18
2170 POKEB,103:POKEB-1024,18    :rem  47
2180 GOSUB9000                  :rem  20
2190 POKEB,108:POKEB-1024,2     :rem 255
2200 GOSUB9000                  :rem  13
2210 POKEB,104:POKEB+1,105      :rem 192
2220 POKEB-1024,8:POKEB-1023,8  :rem 145
2230 POKEB+40,106:POKEB+41,107  :rem 137
2240 POKEB-984,8:POKEB-983,8    :rem  79
2250 NEXT                       :rem   8
2260 FORI=1TOSK*2:GOSUB9000     :rem 136
2270 IFPEEK(B)=102THENGOSUB9000:GOTO2270
                                :rem 152
2280 IFY<13ANDY>7ANDX<5THENGOSUB9000:GOTO
     2270                       :rem  71
2290 POKEB,102:POKEB-1024,7     :rem 255
2300 NEXT                       :rem   4
2310 C=99:X1=1:Y1=10:X2=X1:Y2=Y1:rem  97
2320 RS=0                       :rem 217
2330 POKE3472,109:POKE2448,14   :rem  89
2340 CHAR,2,22,"{BLK}SCORE:":PRINTSC:CHAR
     ,27,22,"TANK:"             :rem  61
2350 FORI=1TOTK:POKE3984+I,97:POKE2960+I,
     0:NEXT                     :rem 126
2360 FORI=9TO11:CHAR,1,I,"{ 3 SPAZI}":NEX
     T                          :rem 137
2490 :                          :rem   9
2500 REM CICLO PRINCIPALE       :rem 252
2510 :                          :rem   2
2520 D$="4"                     :rem 230
2530 SOUND3,98,6                :rem 133
2540 GETZ$:IFZ$<>""THENGOTO3030 :rem  92
2550 Z$=D$:POKEV+X1+Y1*40,C     :rem  20
```

```
2990 :                                :rem  14
3000 REM MOVIMENTO CARRO              :rem 220
3010 :                                :rem 254
3020 IFZ$<>"4"ANDZ$<>"3"ANDZ$<>"W"ANDZ$<>
     "A"THENZ$=D$                     :rem  14
3030 X1=X1+(Z$="A"ANDX1>0)-(Z$="4"ANDX1<3
     9)                               :rem  80
3040 Y1=Y1+(Z$="3"ANDY1>0)-(Z$="W"ANDY1<2
     0)                               :rem  96
3050 C=(97ANDZ$="3")+(98ANDZ$="W")+(99AND
     Z$="4")+(100ANDZ$="A")           :rem  56
3060 B=V+X1+Y1*40:W=PEEK(B)           :rem  27
3070 POKEB,C:POKEB-1024,0:POKEV+X2+Y2*40,
     32                               :rem 238
3080 X2=X1:Y2=Y1                      :rem 167
3090 D$=Z$                            :rem 239
3990 :                                :rem  15
4000 REM CONTROLLO COLLISIONE         :rem  85
4010 :                                :rem 255
4020 IFW<101THEN2530                  :rem 115
4030 IFW=103THENCHAR,0,23,"{FLASH ON}
     [<2>]{ 6 SPAZI}UNA TRAPPOLA ANTICARR
     O!!!{ 8 SPAZI}":GOTO4820         :rem 205
4040 IFW=101THENCHAR,0,23,"{FLASH ON}
     [<2>]{ 9 SPAZI}HAI URTATO UNA MINA!!
     !{ 8 SPAZI}{BLK}":GOTO4820       :rem  44
4050 IFW<>109THEN4090                 :rem 191
4060 IFRS=SK*2THEN6020                :rem  44
4070 X1=1:Y1=10:X2=X1:Y2=Y1:C=99:rem 102
4080 POKE3472,109:POKE2448,14:GOTO2520
                                      :rem 153
4090 IFW<>102THEN4180                 :rem 188
4100 CHAR,5,23,"{FLASH ON}[<2>]POSTAZIONE
      NEMICA DISTRUTTA{BLK}"          :rem 105
4110 FORI=650TO850STEP4               :rem 126
4120 SOUND3,I,1:SOUND3,I+50,1:NEXT
                                      :rem  38
4130 RS=RS+1:SC=SC+10                 :rem 218
4140 CHAR,8,22,"{BLK}":PRINTSC        :rem  44
4150 CHAR,5,23,"{ 30 SPAZI}"          :rem  74
4160 IFRS<SK*2THEN2530                :rem  46
4170 CHAR,1,23,"[<2>]POSTAZIONI DISTRUTTE
      - RIENTRA IN BASE":GOTO2530
                                      :rem 230
4180 CHAR,0,23,"{FLASH ON}[<2>]
     { 7 SPAZI}SEI CADUTO IN UN CRATERE!!
     { 6 SPAZI}"                      :rem 130
4790 :                                :rem  14
4800 REM ESPLOSIONE CARRO             :rem  40
4810 :                                :rem   7
4820 AA=850                           :rem  42
4830 FORI=1TO10                       :rem 113
4840 AA=AA-25:SOUND3,AA,10:POKEB,110
                                      :rem  63
4850 AA=AA-25:SOUND3,AA,6             :rem 172
4860 POKEB,111:NEXT                   :rem 123
4990 :                                :rem  16
5000 REM DISTRUZIONE CARRO            :rem 128
5010 :                                :rem   0
5020 CHAR,0,23,"{FLASH ON}{ 6 SPAZI}IL CA
     RRO E' STATO DISTRUTTO!{ 6 SPAZI}"
                                      :rem 186
5030 POKEB,32                         :rem 202
5040 CHAR,32+TK,22," ":TK=TK-1        :rem  85
5050 X1=1:Y1=10:X2=X1:Y2=Y1:C=99:D$="4"
                                      :rem 188
5060 B=V+X1+Y1*40:POKEB,C:POKEB-1024,0
                                      :rem 172
5070 FORI=1TO1000:NEXT:CHAR,4,23,"
     { 30 SPAZI}"                     :rem   0
```

```
5080 IFTK>0THEN2520                   :rem  97
5190 :                                :rem   9
5200 REM FINE GIOCO                   :rem  62
5210 :                                :rem   2
5220 CHAR,11,23,"{FLASH ON}{BLK}G A M E
     { 3 SPAZI}O V E R"               :rem 222
5230 CHAR,9,24,"GIOCHI ANCORA? (S/N)"
                                      :rem  22
5240 GETZ$:IFZ$="S"THEN2000           :rem  53
5250 IFZ$<>"N"THEN5240                :rem 223
5260 POKE65298,4:POKE65299,208:PRINT"
     {CLR}"                           :rem  70
5270 COLOR0,1:COLOR4,1                :rem  36
5280 CHAR,16,12,"{YEL}C I A O!":END
                                      :rem 107
5990 :                                :rem  17
6000 REM FASE COMPLETATA              :rem 179
6010 :                                :rem   1
6020 CHAR,0,23,"{FLASH ON}{ 10 SPAZI}MISS
     IONE COMPLETATA{ 10 SPAZI}":rem  22
6030 FORI=700TO900STEP4:SOUND1,I,1
                                      :rem  63
6040 SOUND1,I+50,1:NEXT:SC=SC+100
                                      :rem 190
6050 CHAR,8,22,"{ 10 SPAZI}":PRINTSC:FORI
     =1TO100:NEXT                     :rem 179
6060 CHAR,10,23,"{ 24 SPAZI}"         :rem 120
6070 IFSK<>(INT(SK/5)*5)THEN2120:rem  86
6080 CHAR,10,23,"{FLASH ON}[<2>]B O N U S
     { 3 SPAZI}C A R R O"             :rem 143
6090 FORI=500TO900STEP4:SOUND1,I,1:NEXT
                                      :rem 188
6100 TK=TK+1:SC=SC+300                :rem 255
6120 CHAR,8,22," ":PRINTSC            :rem  44
6130 FORI=1TO1000:NEXT                :rem  69
6140 POKE3984+TK,97:POKE2960+TK,0:GOTO212
     0                                :rem 208
8990 :                                :rem  20
9000 REM CALCOLO COORDINATE SCHERMO
                                      :rem 163
9010 :                                :rem   4
9020 X=INT(RND(0)*39)                 :rem 151
9030 Y=INT(RND(0)*20)                 :rem 143
9040 B=V+X+Y*40                       :rem  55
9050 RETURN                           :rem 174
```

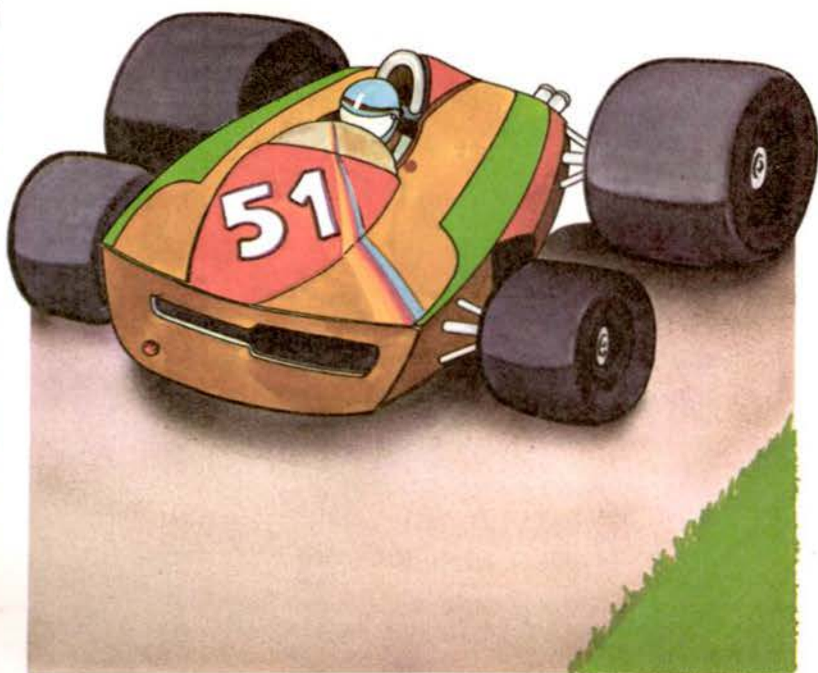




Turbodisk per VIC 20 e C64

di **D. Lewis**
trad. ed adatt.
di **P. Ravizza**

Un'altra super utility! Recentemente, sul numero di Marzo, abbiamo pubblicato il SUPERTURBO per VIC 20 e C64, che permette di caricare i programmi da cassetta alla stessa velocità del disk drive 1541. Ai possessori di quest'ultimo forniamo ora un programma che velocizza del 300% e più il caricamento dei programmi da disco! Vi accorgerete presto che TURBODISK è altrettanto rivoluzionario di SUPERTURBO ed ugualmente facile da usare. Per C64 e per VIC 20 con espansione di memoria da almeno 3 Kbyte.



Se vi è mai capitato di usare un drive realmente veloce, vi sarete accorti che il buon vecchio drive Commodore 1541, in quanto a velocità di caricamento, è uno degli ultimi in classifica. Certamente, è molto più veloce del Datassette senza SUPERTURBO, ma rispetto agli altri drive sembra una tartaruga... Ora esiste una soluzione per portarsi a livelli superiori: TURBODISK!

La prima volta che userete TURBODISK rimarrete sicuramente stupiti dalle sue prestazioni: esso infatti velocizza il processo di caricamento di programmi di un fattore tre o più alto ancora! Infatti, più lungo è il programma da caricare, più velocemente avverrà il suo caricamento.

TURBODISK non richiede alcuna modifica né al vostro computer né al drive: esso carica programmi salvati nella solita maniera; non è richiesto alcun particolare "Turbosave".

TURBODISK è compatibile con la maggior parte dei programmi in BASIC ed in linguaggio macchina, compreso il DOS WEDGE; non compromette in alcun modo l'affidabilità delle operazioni con il drive, e si può attivare o disattivare a piacere, ed in qualunque momento, attraverso due semplici comandi.

Come preparare TURBODISK per C64

Per il C64, come si può notare , ci sono due programmi: il primo – più lungo – è il "programma generatore", che ha il compito di creare un file su disco in linguaggio macchina (il TURBODISK vero e proprio); il secondo è un microprogramma di due linee (non presente per la sua brevità sulla cassetta allegata alla rivista), che avrà in seguito la funzione di caricare ed attivare il TURBODISK.

Ma vediamo, passo dopo passo e per il C64, la procedura da seguire per generare TURBODISK:

1. spegnere e riaccendere il C64, sincerarsi che il drive sia collegato ed acceso;
2. caricare da cassetta il programma "64.TURBODISK" e mandarlo in esecuzione con il RUN;
3. il programma vi chiederà: "INSERISCI IL DISCO E PREMI RETURN". Inutile dire che il disco su cui volete registrare TURBODISK dovrà essere stato precedentemente formattato e deve avere spazio libero sufficiente per contenere il TURBODISK stesso. Inseritelo nel drive, chiudete lo sportello e premete il RETURN;
4. dopo un certo tempo il programma visualizzerà: "TURBODISK.OBJ CREATO" e quindi: "UN'ALTRA COPIA (S/N)?". Premete S, se desiderate creare un'altra copia di TURBODISK: su un altro dischetto, ovviamente, dal momento che uno stesso disco non può contenere due file con lo stesso nome. Premete N in caso negativo.

Tutto qui: molto semplice, no?

Il "generatore" crea un file su disco dal nome "TURBODISK.OBJ": è questo il vero e proprio TURBODISK, che usere-

te per caricare velocemente i programmi da disco. TURBODISK.OBJ è un programma in linguaggio macchina, che parte dalla locazione 49152 e termina nella locazione 50084; la cosa migliore è averne una copia su ognuno dei vostri dischetti: per riprodurlo potete sia mandare in esecuzione ripetutamente il "generatore", cambiando di volta in volta il disco nel drive, sia usare un normale copiatore di file, sia usare il SUPERMONITOR (SUPERCOMMODORE di Gennaio) nella sua versione rilocata a partire dalla locazione 8192.
Infine potete digitare il programmino di due linee qui listato e salvarlo sullo stesso disco, magari con il nome "TURBODISK". La sua unica funzione è quella di caricare automaticamente TURBODISK.OBJ e mandarlo in esecuzione altrettanto automaticamente. Se decidete di non usare questo caricatore, dovrete digitare:

LOAD"TURBODISK.OBJ",8,1

a caricamento avvenuto digitare:

NEW

e quindi digitare:

SYS49152

Il programmino di due linee altro non fa che compiere per voi queste operazioni.
Gli abbonati che non dispongono della cassetta dovranno prestare molta attenzione nel digitare il programma, che comunque è molto ricco di controlli interni sulla correttezza delle linee DATA.

Note alla versione per VIC 20

Il programma per VIC 20 ha un principio di funzionamento diverso da quello per C64. Innanzi tutto occorre sottolineare che è richiesta un'espansione di memoria da almeno 3 Kbyte.
TURBODISK, una volta caricato da cassetta, legge le sue istruzioni DATA e ne inserisce i valori – tramite istruzioni POKE – nella parte alta della memoria.
TURBODISK si regola automaticamente in funzione del tipo di espansione di memoria collegata, così che funziona perfettamente con qualsiasi ammontare di RAM aggiuntiva. TURBODISK ridurrà l'ammontare di memoria libera disponibile di 1280 byte.
TURBODISK può essere salvato su disco come un qualunque programma BASIC (salvate sempre prima di dare il RUN, quindi non caricate il programma con SHIFT+RUN/STOP, ma con un semplice LOAD!).
Una volta mandato in esecuzione il programma visualizzerà due SYS, usabili rispettivamente per attivare e disattivare il TURBODISK; il valore di queste SYS è variabile, a seconda del tipo di espansione di memoria collegata al VIC 20.

Turboloading...

Come già detto, al contrario del SUPERTURBO non occorre ricrearsi il proprio "parco programmi" turbo salvando tutti i file uno per uno: TURBODISK carica velocemente qualsiasi file salvato in modo normale. Dopo aver caricato e mandato in esecuzione TURBODISK.OBJ per il C64, e "VIC.TURBODISK" per il VIC 20, digitate semplicemente:

LOAD"NOME DEL FILE",8
oppure
LOAD"NOME DEL FILE",8,1

proprio come al solito.
Una cosa di cui vi accorgerete immediatamente è che la luce rossa del drive rimane spenta: ciò è assolutamente normale usando TURBODISK. È altrettanto normale, per il C64, lo scomparire dello schermo durante il caricamento. Con il VIC 20 apparirà la scritta "TURBOLOADING..." invece del normale "LOADING".
Potreste, occasionalmente, avere la necessità di disattivare TURBODISK ed usare un normale LOAD: i drive 1541 sono pieni di problemi di allineamento delle testine e se avete un disco formattato da un drive diverso dal vostro, può essere che si creino in questo caso problemi nel caricamento dei programmi da disco. Dal momento che la routine di TURBODISK è poco... insistente nel cercare di caricare programmi che presentano irregolarità di registrazione, sarà più conveniente in questo caso disattivare temporaneamente TURBODISK.
TURBODISK si può disattivare, nel C64, con:

SYS49155

mentre si può riattivare con:

SYS49152

Per il VIC 20 si usino i valori visualizzati dal programma stesso.
Se premete RUN/STOP+RESTORE, dovrete poi riattivare TURBODISK con la SYS di cui sopra.

Alcune precauzioni

Usando TURBODISK, può essere attiva una sola periferica sul bus seriale.
Spegnete quindi tutte le periferiche (quasi sempre stampanti), tranne il drive numero 8. Se "turbocaricate" un programma ed il disco continua a girare indefinitamente, probabilmente avrete dimenticato accesa la stampante.
Nel C64 TURBODISK risiede nel blocco di memoria da 4 Kbyte che parte dalla locazione 49152, in modo da essere completamente sicuro da sovrascritture da parte del BASIC. Tuttavia, alcuni programmi in linguaggio macchina usano il medesimo spazio di memoria occupato da TURBODISK. Ovviamente, questi programmi non sono caricabili in "turbo": occorre disabilitare TURBODISK e procedere ad un caricamento normale; una volta caricato il programma in linguaggio macchina TURBODISK non sarà più presente nella memoria del computer. Ciò vale per tutti quei programmi in linguaggio macchina che occupano le locazioni da 49152 a 50431.
Per quanto riguarda la versione per VIC 20 occorre prestare attenzione nel caricamento di programmi troppo lunghi, che potrebbero sovrascrivere TURBODISK. A questo proposito, dopo aver dato il RUN a TURBODISK stesso digitate in modo diretto:

PRINT INT(FRE(0)/256)

Il valore che si ottiene è – con una certa approssimazione – il massimo numero di blocchi su disco che può occupare un programma per essere caricato senza disturbare TURBODISK.
TURBODISK velocizza il LOAD – anche il LOAD ottenuto da programma, molto comune nella programmazione del VIC 20 – ma lascia inalterati in velocità sia il SAVE che il VERIFY. Non modifica inoltre la velocità della scrittura o caricamento dati dei file sequenziali (istruzioni GET#, INPUT# e PRINT#).

Quanto è veloce?

Ecco i risultati di alcuni test che abbiamo eseguito con il C64, con e senza TURBODISK, sul caricamento di alcuni programmi sia commerciali che da noi precedentemente pubblicati. Vengono riportati i tempi di caricamento con e senza TURBODISK e il fattore di guadagno espresso in percentuale.

PROGRAMMA	BLOCCHI	NORMALE	TURBO	FATTORE
Piccolo database	9	7.0	3.5	200%
Cypher	14	10.5	5.0	210%
Caverne spaziali	17	13.0	5.2	250%
Speedscript	20	14.0	5.3	264%
Forbidden crypt	44	30.0	8.0	375%
Forth apocalypse	153	98.0	23.0	426%

Questi dati sono comunque soggetti a piccole variazioni, a seconda della taratura del drive utilizzato.
Dopo aver provato TURBODISK pensiamo apprezzerete questo logico complemento al SUPERTURBO pubblicato precedentemente.

Non è inoltre compatibile con certe caratteristiche di alcuni programmi, come il caricamento dei testi in SPEEDSCRIPT (anche se è possibile "turbocaricare" SPEEDSCRIPT stesso). Può inoltre non funzionare correttamente con certi programmi commerciali protetti.

Come funziona TURBODISK

Il linguaggio macchina che costituisce TURBODISK è alquanto insolito, dal momento che solo circa la sua metà lavora all'interno del computer, mentre il resto viene eseguito all'interno dello stesso drive 1541. Al contrario di altri drive per diversi computer, il drive Commodore 1541 è una periferica "intelligente", che contiene i propri microprocessori, memoria RAM e memoria ROM.

Ciò significa che il drive può essere programmato per ottenere risultati particolari: TURBODISK è uno di questi risultati.

Durante il breve intervallo di tempo che trascorre dal momento in cui si digita LOAD e quello in cui il disco inizia a girare, vengono trasferiti 420 byte di linguaggio macchina dal computer nella RAM del drive. È questa la seconda parte dei dati nel programma generatore, che nel C64 sono immagazzinati nelle locazioni da 49664 a 50083. Questo necessario trasferimento di dati prima di ogni "turboload" aggiunge un certo ammontare di tempo al tempo totale di caricamento, e spiega il perché TURBODISK sia meno veloce nel caricare programmi brevi e sembri acquistare molta velocità nel caricamento di programmi estremamente lunghi.

TURBODISK modifica il vettore ILOAD nelle locazioni 816-817 ($330-$331) per fare in modo che punti a se stesso, by-passando le normali routine di LOAD nella ROM del computer. Per prima cosa TURBODISK controlla se è stato richiesto un caricamento della directory (LOAD"$"2,8) od un VERIFY, in entrambi i casi il controllo viene ripassato alle routine standard su ROM (noterete infatti che il caricamento della directory avviene a velocità normale). Se invece viene richiesto il caricamento di un programma, TURBODISK aggiunge il nome del file al linguaggio macchina che viene passato al drive ed opera questo passaggio di dati dal computer al drive.

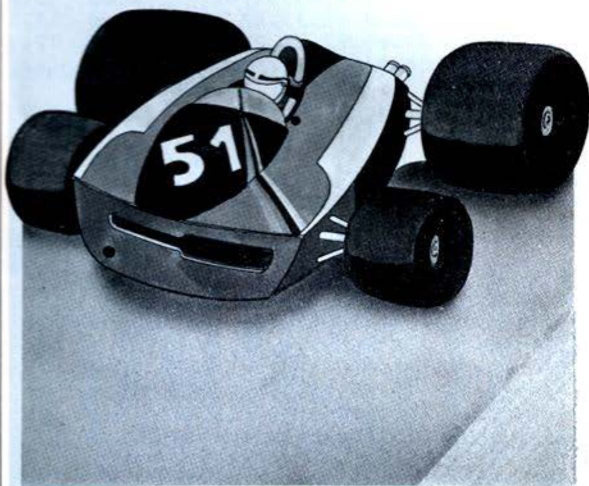
Questa porzione di TURBODISK, che si viene a trovare nella RAM del drive, usa le routine poste invece nella ROM del drive stesso per localizzare la posizione del programma sul disco e quindi caricarlo settore per settore. Per aumentare la velocità di caricamento alcune delle routine ROM del drive – come quella che mantiene accesa la luce rossa – non vengono eseguite. I 256 byte di dati costituenti ogni settore sono trasferiti, due bit alla volta, in un buffer di 256 byte all'interno del computer. Nel C64 questo buffer si trova nelle locazioni 50176-50431($C400-$C4FF).

Il linguaggio macchina costituente TURBODISK all'interno del computer legge i dati in arrivo sulle linee DATA e CLK della porta seriale, invece della sola linea DATA usata nel normale LOAD. In un certo senso TURBODISK trasforma temporaneamente il vostro 1541 seriale in un drive parallelo a due bit. Quando tutti i 256 byte di un settore vengono trasferiti nel buffer del computer il contenuto del buffer viene aggiunto al programma in memoria, mentre il drive sta leggendo dal disco il settore successivo.

TURBODISK per VIC 20

```
10 POKE55,0:POKE56,PEEK(56)-5:CLR:PRINT"
   {GIU'}VIC TURBODISK"                :rem 105
20 X=PEEK(56):A1=X*256:PRINT"{GIU'}SCRITT
   URA BLOCCO 1"                       :rem 160
30 FORAD=A1TOA1+444:READDT:CK=CK+DT:IFDT<
   0THENDT=X-DT-1                      :rem 234
40 POKEAD,DT:NEXT:IFCK<>52477THENPRINT"
   {RVS}ERRORE NELLE":PRINT"LINEE 1000-14
   44":STOP                            :rem  51
50 CK=0:A2=A1+512:PRINT"SCRITTURA BLOCCO
   2"                                  :rem  47
60 FORAD=A2TOA2+419:READDT:CK=CK+DT
                                       :rem 165
70 POKEAD,DT:NEXT:IFCK<>43460THENPRINT"
   {RVS}ERRORE NELLE":PRINT"LINEE 2000-24
   14":STOP                            :rem  45
80 PRINT"DATI OK":PRINT"{GIU'}SYS"A1"PER
   ATTIVARE":PRINT"{GIU'}SYS"A1+3"PER DIS
   ATTIVARE"                           :rem  99
90 SYS A1                              :rem 218
999 REM ** VIC TURBODISK L/M           :rem  68
1000 DATA 24,144,24,169,73,141         :rem 188
1006 DATA 48,3,169,245,141,49          :rem 154
1012 DATA 3,160,0,185,41,-1            :rem  21
1018 DATA 240,6,32,66,231,200          :rem 133
1024 DATA 208,245,96,169,84,141        :rem   3
1030 DATA 48,3,169,-1,141,49           :rem  90
1036 DATA 3,160,21,208,230,13          :rem 128
1042 DATA 84,85,82,66,79,68            :rem  74
1048 DATA 73,83,75,32,68,73            :rem  65
1054 DATA 83,65,66,76,69,68            :rem  76
1060 DATA 13,0,13,84,85,82             :rem 248
1066 DATA 66,79,68,73,83,75            :rem  78
1072 DATA 32,65,67,84,73,86            :rem  65
1078 DATA 65,84,69,68,13,0             :rem  14
1084 DATA 133,147,165,147,208,30:rem  43
1090 DATA 160,0,177,187,201,36         :rem 194
1096 DATA 240,22,162,16,169,160        :rem 247
1102 DATA 157,148,-4,202,16,250        :rem 231
1108 DATA 177,187,153,148,-4,200:rem  40
1114 DATA 196,183,144,246,176,5        :rem   1
1120 DATA 165,147,76,73,245,160        :rem 251
1126 DATA 0,185,138,-1,240,30          :rem 129
1132 DATA 32,66,231,200;208,245        :rem 235
1138 DATA 13,84,85,82,66,79            :rem  70
1144 DATA 76,79,65,68,73,78            :rem  79
1150 DATA 71,46,46,46,13,0             :rem 247
1156 DATA 77,45,87,0,0,32              :rem 206
1162 DATA 169,16,133,255,169,0         :rem 202
1168 DATA 133,251,169,-3,133,252:rem  37
1174 DATA 169,0,133,253,169,5          :rem 153
1180 DATA 133,254,165,186,32,177:rem  49
1186 DATA 255,169,111,32,147,255:rem  50
1192 DATA 165,253,164,254,141,159
                                       :rem 100
1198 DATA -1,140,160,-1,160,0          :rem 120
1204 DATA 185,156,-1,32,168,255        :rem 245
1210 DATA 200,192,6,208,245,160        :rem 234
1216 DATA 0,177,251,32,168,255         :rem 199
1222 DATA 200,192,32,144,246,165:rem  33
1228 DATA 251,105,31,133,251,165:rem  32
1234 DATA 252,105,0,133,252,165        :rem 235
1240 DATA 253,105,32,133,253,165:rem  31
1246 DATA 254,105,0,133,254,32         :rem 187
1252 DATA 174,255,198,255,208,180
                                       :rem 105
1258 DATA 165,186,32,177,255,169:rem  65
1264 DATA 111,32,147,255,169,85        :rem   0
1270 DATA 32,168,255,169,67,32         :rem 210
1276 DATA 168,255,32,174,255,120:rem  49
1282 DATA 32,137,-2,44,0,-5            :rem  28
1288 DATA 48,59,164,195,166,196        :rem  27
1294 DATA 165,185,240,6,172,2          :rem 152
1300 DATA -5,174,3,-5,132,174          :rem 127
1306 DATA 134,175,162,4,32,113         :rem 188
1312 DATA -2,32,137,-2,173,0           :rem  70
1318 DATA -5,48,41,240,6,32            :rem  39
1324 DATA 111,-2,24,144,240,162        :rem 224
1330 DATA 2,160,0,189,0,-5             :rem 234
1336 DATA 145,174,200,232,240,7        :rem 239
```

```
1342 DATA 236,1,-5,144,242,240   :rem  181
1348 DATA 240,32,124,-2,24,166   :rem  186
1354 DATA 174,164,175,88,96,169  :rem   22
1360 DATA 4,44,169,0,56,176      :rem   53
1366 DATA 242,162,2,160,0,189    :rem  146
1372 DATA 0,-5,145,174,200,232   :rem  180
1378 DATA 208,247,24,152,101,174:rem   44
1384 DATA 133,174,165,175,105,0  :rem  248
1390 DATA 133,175,96,160,0,169   :rem  205
1396 DATA 128,141,17,145,173,17  :rem  255
1402 DATA 145,41,2,240,249,169   :rem  195
1408 DATA 0,141,17,145,162,7     :rem   91
1414 DATA 202,234,208,252,162,4  :rem  237
1420 DATA 173,17,145,74,38,149   :rem  206
1426 DATA 74,38,149,234,234,202  :rem  253
1432 DATA 208,242,165,149,73,255:rem   50
1438 DATA 153,0,-5,200,208,207   :rem  182
1444 DATA 96                     :rem   86
1999 REM ** 1541 TURBODISK L/M   :rem   94
2000 DATA 32,66,208,120,169,18   :rem  191
2006 DATA 160,1,141,0,3,140      :rem   20
2012 DATA 1,3,32,186,5,169       :rem  248
2018 DATA 3,133,60,162,0,134     :rem   82
2024 DATA 75,240,41,160,0,177    :rem  139
2030 DATA 59,201,130,208,25,200  :rem  227
2036 DATA 200,200,185,145,6,201  :rem  230
2042 DATA 42,240,61,201,63,240   :rem  179
2048 DATA 4,209,59,208,7,200     :rem  100
2054 DATA 192,18,240,48,208,234  :rem  251
2060 DATA 230,75,166,75,224,8    :rem  152
2066 DATA 240,7,189,98,5,133     :rem  112
2072 DATA 59,208,208,173,0,3     :rem   97
2078 DATA 240,6,172,1,3,76       :rem  254
2084 DATA 14,5,169,255,141,0     :rem   96
2090 DATA 3,32,133,5,88,76       :rem    2
2096 DATA 69,217,2,34,66,98      :rem   70
2102 DATA 130,162,194,226,230,59:rem   35
2108 DATA 160,0,177,59,141,0     :rem   91
2114 DATA 3,200,177,59,141,1     :rem   87
2120 DATA 3,32,186,5,32,133      :rem   35
2126 DATA 5,173,0,3,208,245      :rem   41
2132 DATA 96,160,0,185,0,3       :rem  245
2138 DATA 133,133,44,0,24,16     :rem   87
2144 DATA 251,169,16,141,0,24    :rem  140
2150 DATA 44,0,24,48,251,162     :rem   89
2156 DATA 4,169,0,6,133,42       :rem  251
2162 DATA 10,6,133,42,10,141     :rem   76
2168 DATA 0,24,202,208,240,72    :rem  138
2174 DATA 104,72,104,169,15,141  :rem  243
2180 DATA 0,24,200,208,206,96    :rem  138
2186 DATA 172,1,3,132,7,173      :rem   45
2192 DATA 0,3,197,6,8,133        :rem  205
2198 DATA 6,40,240,16,169,176    :rem  159
2204 DATA 133,0,88,36,0,48       :rem  250
2210 DATA 252,120,165,0,201,1    :rem  119
2216 DATA 208,78,169,238,141,12  :rem    0
2222 DATA 28,169,6,133,50,169    :rem  154
2228 DATA 0,133,51,133,48,169    :rem  148
2234 DATA 3,133,49,32,58,6       :rem    0
2240 DATA 80,254,184,173,1,28    :rem  148
2246 DATA 153,0,3,200,208,244    :rem  134
2252 DATA 160,186,80,254,184,173:rem   50
2258 DATA 1,28,153,0,1,200       :rem  238
2264 DATA 208,244,32,224,248,165:rem   47
2270 DATA 56,197,71,240,4,169    :rem  159
2276 DATA 34,208,20,32,233,245   :rem  194
2282 DATA 197,58,240,4,169,35    :rem  164
2288 DATA 208,9,169,236,141,12   :rem  209
2294 DATA 28,96,24,105,24,133    :rem  153
2300 DATA 68,169,255,141,0,3     :rem   94
2306 DATA 32,133,5,165,68,76     :rem  105
2312 DATA 200,193,32,64,6,76     :rem   95
2318 DATA 124,6,165,18,133,22    :rem  145
2324 DATA 165,19,133,23,165,6    :rem  149
2330 DATA 133,24,165,7,133,25    :rem  140
2336 DATA 169,0,69,22,69,23      :rem   59
2342 DATA 69,24,69,25,133,26     :rem  107
2348 DATA 32,52,249,162,90,32    :rem  153
2354 DATA 124,6,80,254,184,173   :rem  204
2360 DATA 1,28,217,36,0,208      :rem   41
2366 DATA 6,200,192,8,208,240    :rem  147
2372 DATA 96,202,208,233,169,32  :rem  254
2378 DATA 208,175,169,208,141,5  :rem    6
2384 DATA 24,169,33,44,5,24      :rem   54
2390 DATA 16,163,44,0,28,48      :rem   51
2396 DATA 246,173,1,28,184,160   :rem  208
2402 DATA 0,96,160,160,160,160   :rem  185
2408 DATA 160,160,160,160,160,160
                                 :rem   78
2414 DATA 160,160,160,160,160,160
                                 :rem   75
```



Generatore TURBODISK per C64

```
100 PRINT"{CLR}"TAB(206)"{WHT}GENERATORE
    TURBODISK":PRINT:PRINT          :rem   47
110 PRINT"{CYN}INSERISCI IL DISCO E PREMI
     {RVS}RETURN{OFF}":PRINT:PRINT
                                    :rem  134
120 GETA$:IFA$<>CHR$(13)THEN120     :rem  248
130 OPEN2,8,2,"TURBODISK.OBJ,P,W":GOSUB 1
    000                             :rem  100
140 PRINT#2,CHR$(0)CHR$(192);       :rem   78
150 FORI=0TO427:READA:CK=CK+A:PRINT#2,CHR
    $(A);:NEXTI                     :rem  225
160 IFA<>32ORCK<>55038THENPRINT"{RVS}ERRO
    RE NELLE LINEE DATA 49152-49578":GOTO
    300                             :rem   21
170 FORI=0TO83:PRINT#2,CHR$(234);:NEXTI
                                    :rem  115
180 CK=0:FORI=0TO419:READA:CK=CK+A:PRINT#
    2,CHR$(A);:NEXTI                :rem   26
190 IFA<>160ORCK<>43460THENPRINT"{RVS}ERR
    ORE NELLE LINEE DATA 49664-50078":GOT
    O300                            :rem   65
```

```
200 CLOSE 2:PRINTTAB(9)"[<7>]TURBODISK.OB
    J CREATO":PRINT:PRINTTAB(10);
                                :rem  16
210 INPUT "UN'ALTRA COPIA (S/N)";A$:IFA$<
    >"S" THEN END               :rem  23
220 RUN                         :rem 137
300 CLOSE2:CLOSE15:OPEN15,8,15,"S0:TURBOD
    ISK.OBJ":CLOSE15:END        :rem  45
1000 CLOSE15:OPEN15,8,15:INPUT#15,E,E$,T,
    S:IFE=0THENRETURN           :rem  71
1010 PRINT"ERRORE DISCO"E": "E$;T;S
                                :rem  29
1020 CLOSE15:OPEN15,8,15,"I0:":CLOSE15
                                :rem 160
49100 REM ** 64 TURBODISK L/M   :rem  31
49152 DATA 76,27,192,169,165,141:rem  63
49158 DATA 48,3,169,244,141,49  :rem 221
49164 DATA 3,160,0,185,41,192   :rem 151
49170 DATA 240,6,32,22,231,200  :rem 184
49176 DATA 208,245,96,169,84,141:rem  71
49182 DATA 48,3,169,192,141,49  :rem 220
49188 DATA 3,160,21,208,230,13  :rem 196
49194 DATA 84,85,82,66,79,68    :rem 142
49200 DATA 73,83,75,32,68,73    :rem 115
49206 DATA 83,65,66,76,69,68    :rem 135
49212 DATA 13,0,13,84,85,82     :rem  51
49218 DATA 66,79,68,73,83,75    :rem 137
49224 DATA 32,65,67,84,73,86    :rem 124
49230 DATA 65,84,69,68,13,0     :rem  64
49236 DATA 133,147,165,147,208,30
                                :rem 102
49242 DATA 162,16,169,160,157,148
                                :rem 108
49248 DATA 195,202,16,250,160,0 :rem 249
49254 DATA 177,187,201,36,240,12:rem  50
49260 DATA 177,187,153,148,195,200
                                :rem 161
49266 DATA 196,183,144,246,176,5:rem  69
49272 DATA 165,147,76,165,244,32:rem  62
49278 DATA 69,193,165,186,32,177:rem  79
49284 DATA 255,169,111,32,147,255
                                :rem 109
49290 DATA 169,85,32,168,255,169:rem  75
49296 DATA 67,32,168,255,32,174 :rem  18
49302 DATA 255,120,169,11,141,17:rem  39
49308 DATA 208,32,19,193,44,0   :rem 156
49314 DATA 196,48,76,164,195,166:rem  75
49320 DATA 196,165,185,240,6,172:rem  56
49326 DATA 2,196,174,3,196,132  :rem 213
49332 DATA 174,134,175,162,4,32 :rem 254
49338 DATA 251,192,32,19,193,173:rem  60
49344 DATA 0,196,48,48,240,6    :rem 114
49350 DATA 32,249,192,76,188,192:rem  69
49356 DATA 162,2,160,0,189,0    :rem 101
49362 DATA 196,145,174,200,232,240
                                :rem 147
49368 DATA 7,236,1,196,144,242  :rem 216
49374 DATA 240,240,32,6,193,24  :rem 203
49380 DATA 72,169,27,141,17,208 :rem   8
49386 DATA 104,166,174,164,175,88
                                :rem 122
49392 DATA 96,169,4,44,169,0    :rem 124
49398 DATA 56,176,235,162,2,160 :rem  12
49404 DATA 0,189,0,196,145,174  :rem 211
49410 DATA 200,232,208,247,24,152
                                :rem  86
49416 DATA 101,174,133,174,165,175
                                :rem 152
49422 DATA 105,0,133,175,96,160 :rem 251
49428 DATA 0,169,11,141,0,221   :rem 142
49434 DATA 173,0,221,16,251,169 :rem 253
49440 DATA 3,141,0,221,162,5    :rem  87
49446 DATA 202,234,208,252,162,4:rem  46
49452 DATA 173,0,221,10,8,10    :rem  88
49458 DATA 38,149,40,38,149,202 :rem  14
49464 DATA 208,242,165,149,73,255
                                :rem 115
49470 DATA 153,0,196,200,208,209:rem  46
49476 DATA 96,169,16,133,255,169:rem  77
49482 DATA 0,133,251,169,194,133:rem  53
49488 DATA 252,169,0,133,253,169:rem  65
49494 DATA 5,133,254,165,186,32 :rem  11
49500 DATA 177,255,169,111,32,147
                                :rem 103
49506 DATA 255,165,253,164,254,141
                                :rem 156
49512 DATA 169,193,140,170,193,160
                                :rem 153
49518 DATA 0,185,166,193,32,168 :rem  13
49524 DATA 255,200,192,6,208,245:rem  51
49530 DATA 160,0,177,251,32,168 :rem 253
49536 DATA 255,200,192,32,144,246
                                :rem 101
49542 DATA 165,251,105,31,133,251
                                :rem  91
49548 DATA 165,252,105,0,133,252:rem  47
49554 DATA 165,253,105,32,133,253
                                :rem  99
49560 DATA 165,254,105,0,133,254:rem  45
49566 DATA 32,174,255,198,255,208
                                :rem 121
49572 DATA 180,96,77,45,87,0    :rem 127
49578 DATA 0,32                 :rem 236
49600 REM ** 1541 TURBODISK L/M :rem 133
49664 DATA 32,66,208,120,169,18 :rem  10
49670 DATA 160,1,141,0,3,140    :rem  86
49676 DATA 1,3,32,186,5,169     :rem  67
49682 DATA 3,133,60,162,0,134   :rem 148
49688 DATA 75,240,41,160,0,177  :rem 214
49694 DATA 59,201,130,208,25,200:rem  46
49700 DATA 200,200,185,145,6,201:rem  31
49706 DATA 42,240,61,201,63,240 :rem 245
49712 DATA 4,209,59,208,7,200   :rem 157
49718 DATA 192,18,240,48,208,234:rem  61
49724 DATA 230,75,166,75,224,8  :rem 218
49730 DATA 240,7,189,98,5,133   :rem 169
49736 DATA 59,208,208,173,0,3   :rem 163
49742 DATA 240,6,172,1,3,76     :rem  55
49748 DATA 14,5,169,255,141,0   :rem 162
49754 DATA 3,32,133,5,88,76     :rem  68
49760 DATA 69,217,2,34,66,98    :rem 127
49766 DATA 130,162,194,226,230,59
                                :rem 110
49772 DATA 160,0,177,59,141,0   :rem 157
49778 DATA 3,200,177,59,141,1   :rem 162
49784 DATA 3,32,186,5,32,133    :rem 110
49790 DATA 5,173,0,3,208,245    :rem 107
49796 DATA 96,160,0,185,0,3     :rem  64
49802 DATA 133,133,44,0,24,16   :rem 144
49808 DATA 251,169,16,141,0,24  :rem 206
49814 DATA 44,0,24,48,251,162   :rem 155
49820 DATA 4,169,0,6,133,42     :rem  52
49826 DATA 10,6,133,42,10,141   :rem 142
49832 DATA 0,24,202,208,240,72  :rem 195
49838 DATA 104,72,104,169,15,141:rem  53
49844 DATA 0,24,200,208,206,96  :rem 204
49850 DATA 172,1,3,132,7,173    :rem 102
49856 DATA 0,3,197,6,8,133      :rem  15
49862 DATA 6,40,240,16,169,176  :rem 216
49868 DATA 133,0,88,36,0,48     :rem  69
49874 DATA 252,120,165,0,201,1  :rem 194
49880 DATA 208,78,169,238,141,12:rem  66
```

```
49886 DATA 28,169,6,133,50,169    :rem  229
49892 DATA 0,133,51,133,48,169    :rem  214
49898 DATA 3,133,49,32,58,6       :rem   75
49904 DATA 80,254,184,173,1,28    :rem  214
49910 DATA 153,0,3,200,208,244    :rem  191
49916 DATA 160,186,80,254,184,173
                                  :rem  116
49922 DATA 1,28,153,0,1,200       :rem   39
49928 DATA 208,244,32,224,248,165
                                  :rem  113
49934 DATA 56,197,71,240,4,169    :rem  225
49940 DATA 34,208,20,32,233,245   :rem  251
49946 DATA 197,58,240,4,169,35    :rem  230
49952 DATA 208,9,169,236,141,12   :rem   10
49958 DATA 28,96,24,105,24,133    :rem  219
49964 DATA 68,169,255,141,0,3     :rem  169
49970 DATA 32,133,5,165,68,76     :rem  171
49976 DATA 200,193,32,64,6,76     :rem  170
49982 DATA 124,6,165,18,133,22    :rem  211
49988 DATA 165,19,133,23,165,6    :rem  224
49994 DATA 133,24,165,7,133,25    :rem  215
50000 DATA 169,0,69,22,69,23      :rem   98
50006 DATA 69,24,69,25,133,26     :rem  155
50012 DATA 32,52,249,162,90,32    :rem  192
50018 DATA 124,6,80,254,184,173   :rem  252
50024 DATA 1,28,217,36,0,208      :rem   89
50030 DATA 6,200,192,8,208,240    :rem  186
50036 DATA 96,202,208,233,169,32:rem   46
50042 DATA 208,175,169,208,141,5:rem   45
50048 DATA 24,169,33,44,5,24      :rem  102
50054 DATA 16,163,44,0,28,48      :rem   99
50060 DATA 246,173,1,28,184,160   :rem  247
50066 DATA 0,96,160,160,160,160   :rem  242
```




```
50072 DATA 160,160,160,160,160,160
                                  :rem  126
50078 DATA 160,160,160,160,160,160
                                  :rem  132
```

Caricatore TURBODISK per C64

```
10 IF A=0THENA=1:LOAD "TURBODISK.OBJ",8,1
                                  :rem  155
20 SYS 49152:NEW                  :rem  138
```

MISSIONE AFRICA

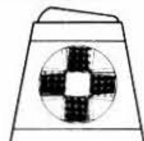
Una "Nave della Pace" in partenza dall'Italia porterà in Africa soccorsi immediati e aiuti per un domani migliore con il vostro contributo.



La "Nave della Pace": uno strumento nuovo per intervenire immediatamente là dove è necessario un urgente soccorso a popolazioni gravemente colpite da calamità naturali. Non porta solo aiuti immediati ma anche strumenti di lavoro e mezzi ausiliari idonei a realizzare migliori condizioni di vita.



Affinché questo risultato sia raggiunto, gli aiuti siano adeguati ed efficaci e la "Nave della Pace" possa partire a pieno carico occorrono offerte e solidarietà da parte di tutti. Specialisti dei problemi dello sviluppo hanno offerto consigli per la migliore attuazione del progetto. L'iniziativa ha ottenuto l'Alto Patronato del Presidente della Repubblica Italiana, il Patrocinio dell'ONU, della Lega di Società di Croce Rossa e di Mezza Luna Rossa e della Croce Rossa Italiana e il contributo del Dipartimento per la Cooperazione allo Sviluppo del Ministero degli Affari Esteri. Il Comitato "Nave della Pace" chiede a quanti comprendono la necessità ed il valore dell'iniziativa, offerte di mezzi e di beni indispensabili per i soccorsi di prima necessità e per gli aiuti di sviluppo destinati a 15 paesi dell'Africa.



Nave della Pace

COMITATO NAVE DELLA PACE - VIALE MAZZINI 41
00195 ROMA - TEL. 06/317447-386163

Da compilare in stampatello ed inviare in busta chiusa a:
Comitato Nave della Pace
V.le Mazzini 41 - 00195 Roma.

SI', ANCH'IO VOGLIO AIUTARE LA NAVE DELLA PACE A PARTIRE CON LE STIVE PIENE

Cognome__________
Nome__________
Via__________ **N.**_____
CAP_____ **Località**__________

Per questo ho deciso di inviare il mio contributo di

- ☐ **Lit. 10.000** ☐ **Lit. 50.000**
- ☐ **Lit. 25.000** ☐ **Lit. 100.000 o più**

tramite:

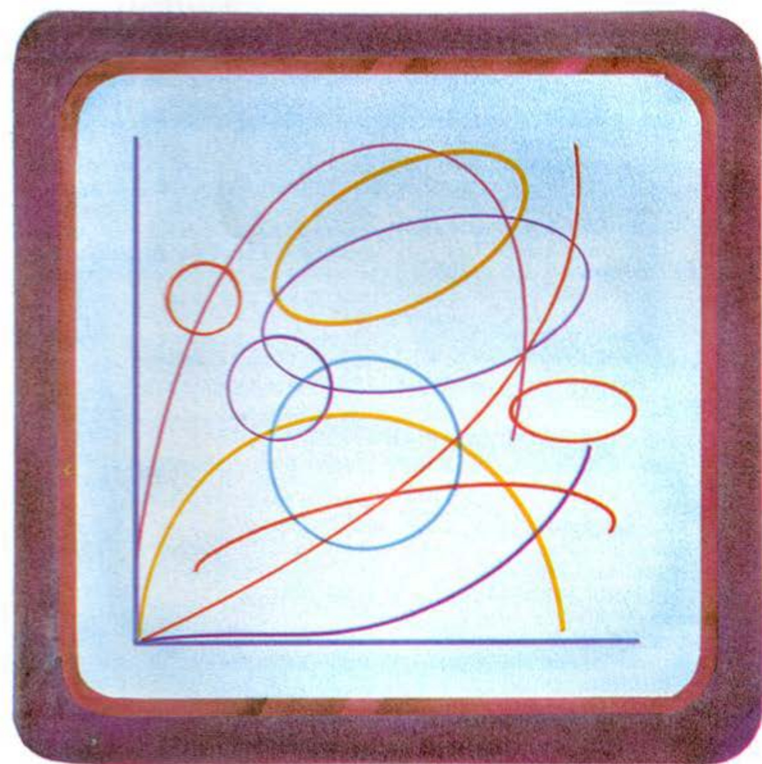
- ☐ **Assegno non trasferibile intestato: Comitato Nave della Pace**
- ☐ **C/c postale n. 15285000**
- ☐ **Bonifico bancario a credito del c/c n. 3100/51 c/o la Cassa di Risparmio di Roma sede centrale - Via del Corso, 320 - 00186 Roma**
- ☐ **Desidero una ricevuta del mio versamento.**



Conic curve plotter per C64

di **L. Wong**
trad. ed adatt.
di **L. Priotto**

Il C64 è un computer potente e versatile, ma il suo interprete BASIC non è dotato di apposite istruzioni per tracciare grafici in alta risoluzione. Ecco quindi un programma che rende facilissimo l'uso della pagina grafica, fornendovi di un nutrito set di comandi per la visualizzazione di una gran varietà di figure geometriche.



"Conic curve plotter" permette di disegnare linee, angoli, cerchi, ellissi, parabole ed iperboli sullo schermo ad alta risoluzione del C64 con la semplice pressione di un tasto. Permette inoltre di disegnare con il joystick e di salvare la pagina grafica su nastro.
Lo schermo ad alta risoluzione del C64, altrimenti detto pagina grafica o schermo a mappa di bit, è dotato di una risoluzione di 320 pixel orizzontali per 200 pixel verticali.
Nelle versioni "sperimentali" di questo programma i cerchi apparivano alquanto schiacciati, proprio a causa della forma rettangolare dei pixel. Dopo aver eseguito un po' di calcoli con la carta millimetrata abbiamo scoperto che la lunghezza di 9 pixel in verticale equivale a una larghezza di 11 pixel in orizzontale; per rimediare a questo inconveniente insito, nell'hardware del C64, abbiamo definito all'interno del programma ben tre tipi diversi di schermo.
Il primo è il normale schermo in alta risoluzione, con 320 pixel verticali e 200 orizzontali; il secondo ha dimensioni di 320 pixel orizzontali per 244 verticali (risultato ottenuto moltiplicando la normale ordinata verticale per 11/9). Il terzo schermo permette che le sue dimensioni vengano definite dall'utente. Naturalmente, le dimensioni citate della pagina grafica sono "fittizie", usate cioè nei calcoli di scala effettuati dal programma per poter generare figure non distorte, e non costituiscono in alcun modo un aumento di risoluzione del C64.
Il secondo schermo è quello che minimizza le distorsioni nei cerchi e negli angoli, ed è quindi quello consigliato normalmente, mentre il terzo può servire per particolari esperimenti di grafica.
I confini dello schermo vengono controllati dal programma in tutti i modi di tracciamento, con l'esclusione di quello tramite joystick. Il tracciamento di linee, parabole ed iperboli può essere interrotto manualmente, e si interrompe in ogni caso, se vengono raggiunti i confini dello schermo.
Tracciando cerchi, archi o ellissi non è invece possibile interrompere manualmente il procedimento di visualizzazione.
In origine il programma era stato scritto interamente in BASIC: ci volevano ben 27 secondi per la pulizia della pagina grafica (inserendo degli 0 nelle locazioni da 8192 a 16191), 3 secondi per regolare il suo colore di fondo sul ciano (inserendo il valore 3 nelle locazioni da 1024 a 2023) e 25 minuti per salvare o caricare la schermata su nastro (con l'uso di PRINT # e GET #). In seguito queste routine molto lente in BASIC sono state sostituite con altre routine in linguaggio macchina, ottenendo funzioni

praticamente istantanee e riducendo il tempo di salvataggio delle schermate su nastro a circa quattro minuti.

Inizializzare il tutto

All'inizio dell'esecuzione del programma vi verrà richiesto di selezionare il tipo di schermo. Dopo che avrete inserito la vostra scelta si passerà in alta risoluzione ed il cursorino (un pixel nero) apparirà al centro dello schermo.
Siete ora pronti per tracciare un disegno, usando i comandi del programma.
Molti di essi richiedono l'inserimento di informazioni addizionali, come una posizione di schermo od un angolo. La posizione sullo schermo viene definita inserendo le coordinate X e Y, con origine in alto a sinistra, di coordinate quindi 0,0. Dal momento che il programma non controlla la congruenza dei valori di coordinate inseriti, assicuratevi di inserire valori che non superino le dimensioni dello schermo che avete precedentemente selezionato.
Gli angoli devono sempre essere specificati in gradi – non in radianti – e misurati in senso orario a partire dall'orizzontale.

Comandi con la pressione di un tasto

A: traccia una linea con angolo specificato. Vi viene richiesto il punto di partenza e l'angolo della linea. Il cursore inizierà a muoversi tracciando la linea, fino a che venga raggiunto il bordo dello schermo, o venga premuto un tasto qualsiasi.

O: traccia una linea tra due punti. Vi viene richiesto il punto di partenza ed il punto di arrivo. Il cursore può essere fermato premendo il tasto F1. La pressione di F3 seleziona invece i due modi "draw" ed "erase" ed inverte la direzione del cursore.

C: traccia un cerchio od un arco. Occorre specificarne il raggio, il centro, l'angolo di partenza e la densità. Per tracciare un cerchio gli angoli iniziale e finale sono rispettivamente 0 e 360. La densità può assumere valori tra 0.1 e 1, premendo RETURN si assume il valore di default di 0.667. La densità determina la spaziatura tra i pixel plottati: un valore basso causerà un cerchio composto da pochi punti, mentre un valore alto traccerà un cerchio molto più definito, impiegando tuttavia un tempo superiore.

I: traccia un'ellisse. Occorre specificare i parametri A e B dell'equazione (X*X/A*A+Y*/B*B=1), il centro, l'angolo dell'asse maggiore e la densità. Il parametro A è metà della lunghezza dell'asse maggiore, mentre B è la metà dell'asse minore. Se A e B sono uguali, verrà tracciato un cerchio di raggio A.

P: traccia una parabola. Vi verrà richiesto il parametro A dell'equazione (X=A*Y*Y), il vertice, l'angolo dell'asse di simmetria e la densità. Il tracciamento può essere fermato in qualunque momento con la pressione di un tasto.

H: traccia un'iperbole. Occorre specificare i parametri A e B dell'equazione (X*X/A*A−Y*Y/B*B=1), il centro, l'angolo dell'asse trasversale e la densità. Anche qui è possibile fermare il tracciamento con la pressione di un tasto qualunque.

Q: permette di definire la posizione del cursore, il tipo di schermo e le sue dimensioni.

T: modifica il tipo di schermo. Questo comando viene eseguito automaticamente all'inizio del programma. I disegni tracciati precedentemente non vengono disturbati dal combiamento del tipo di schermo.

M: muove il cursore a un punto specificato. Il tasto CLR/HOME porta il cursore all'origine delle coordinate, mentre SHIFT+CLR/HOME pulisce anche la pagina grafica.

S: salva la pagina grafica su nastro.

L: carica una pagina grafica precedentemente salvata su nastro.

/: esce dal programma. Per rientrare nel programma, senza perdere i disegni tracciati, digitate GOTO15.

Il programma è diviso in due parti. La prima inserisce in memoria il linguaggio macchina, mentre la seconda rappresenta il programma principale. Occorre ovviamente caricare la prima parte, dare il RUN, e alla ricomparsa del cursore caricare la seconda parte e darle a sua volta il RUN.

Un esempio

A titolo di esempio vi forniamo i parametri usati per tracciare la schermata del fotocolor

TASTO DA PREMERE	PARAMETRI R	A	B	ANGOLI INIZ.	FIN.	TRA ASSI	COORD. CENTRO	DENSITÀ PUNTI
C	75	–	–	0	360	–	160,123	1.0
C	30	–	–	0	360	–	160,123	1.0
I	–	75	40	–	–	90	160,123	0.667
I	–	75	40	–	–	0	160,123	0.667
I	–	20	13	–	–	135	80,35	1.0
I	–	20	13	–	–	45	80,210	1.0
I	–	20	13	–	–	135	239,210	1.0
I	–	20	13	–	–	45	239,35	1.0

Conic curve plotter –
Programma 1

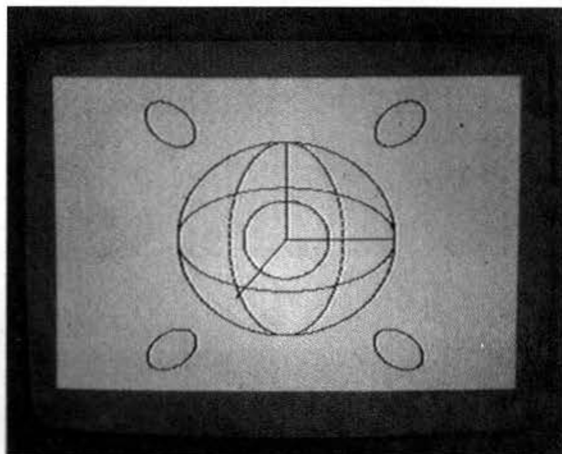
```
10 FORI=49360TO49461:READJ:POKEI,J:NEXT
                                 :rem 192
20 FORI=49485TO49623:READJ:POKEI,J:NEXT
                                 :rem 201
30 DATA 173,0,192,133,0,173,2,192,133,2,1
   73,3,192,133,3,96,165,3,141  :rem 253
40 DATA 3,192,165,2,141,2,192,165,0,141,0
   ,192,96,0,165,3,141,3,192,165:rem  96
50 DATA 2,141,2,192,165,0,32,224,192,169,
   32,133,3,169,0,133,2,133     :rem 101
60 DATA 0,164,0,162,0,145,2,230,2,232,224
   ,255,208,247,145,2,230,3,169 :rem  37
70 DATA 63,197,3,208,227,169,0,162,0,145,
   2,230,2,232,224,63,208,247,145
                                 :rem 160
80 DATA 2,141,63,63,76,208,192  :rem 199
90 DATA 32,224,192,169,59,141,17,208,169,
   28,141,24                    :rem 172
100 DATA 208,169,4,133,3,169,0,133,2,133,
    0,162,0,164,0,169,3,145,2,230
                                :rem  79
110 DATA 2,232,224,255,208,247,145,2,230,
    3,169,7,197,3,208,225,169,3 :rem  17
120 DATA 162,0,145,2,230,2,232,224,231,20
    8,247,145,2,141,231,7,76,208,192
                                :rem 239
130 DATA 0,0,0,0,32,224,192,160,255,162,1
    ,169,1,32,186,255,169,0     :rem  49
140 DATA 32,189,255,169,0,133,2,169,32,13
    3,3,162,64,160,63           :rem  37
150 DATA 169,2,32,216,255,76,208,192,0,0,
    0,0,0                       :rem 191
160 DATA 169,1,162,1,160,255,32,186,255,1
    69,0,32,189,255             :rem 206
170 DATA 169,0,162,0,160,32,32,213,255,96
                                :rem 217
```

Conic curve plotter – Programma 2

```
14 SYS 49405                              :rem 106
15 POKE46,71:CLR:BASE=8192                :rem  20
16 X=160:Y=100:N=1:MODE$="DRAW":CP=53272:
   BM=53265:BO=PEEK(53265)AND223:rem       33
17 POKECP,21:POKEBM,BO:PRINT"{CLR}"
                                          :rem  12
18 DIMX(11):DIMY(11):FORK=0TO10:READX(K),
   Y(K):NEXT                              :rem  47
19 DATA 0,0,0,-1,0,1,0,0,-1,0,-1,-1,-1,1,
   0,0,1,0,1,-1,1,1                       :rem  90
20 PRINTSPC(12)"{RVS}TIPO DI SCHERMO:":PR
   INT                                    :rem 186
21 PRINT"1-'SCALA NORMALE' (320,200)":PRI
   NT"2-'SCALA SPECIALE' (320,244)"
                                          :rem   4
22 PRINT"3 -'SCALA DEFINITA DALL'UTENTE'"
                                          :rem 164
24 PRINT:INPUTSCR:IFSCR=1THENXP=1:YP=1:GO
   TO30                                   :rem  48
26 IFSCR=2THENXP=1:YP=9/11:GOTO30
                                          :rem  83
27 IFSCR=3THENPRINT:INPUT"DIMENSIONI (X,Y
   ):";XM,YM:IFXM=0ORYM=0THEN27 :rem       74
28 XP=320/XM:YP=200/YM:GOTO35             :rem  26
29 GOTO24                                 :rem  10
30 XM=320:YM=INT(200/YP)                  :rem 157
35 SYS 49485                              :rem 117
40 DEF FN FY(Y)=INT(Y/8)*320+(YAND7):DEF
   FN FX(X)=8*INT(X/8)                    :rem 101
42 DEF FN B1(O)=FNFY(Y)+FNFX(X)+8192:DEF
   FN B2(X)=7-(7ANDX)                     :rem   9
45 BY=FNB1(0):BI=FNB2(X):POKEBY,PEEK(BY)O
   R(2↑BI)                                :rem 202
50 GETF$                                  :rem 175
60 IFF$="J"THEN90                         :rem 194
61 IFF$="O"THEN1000                       :rem  32
62 IFF$="C"THEN1300                       :rem  24
63 IFF$="I"THEN1320                       :rem  33
64 IFF$="P"THEN800                        :rem 251
65 IFF$="H"THEN900                        :rem 245
66 IFF$="Q"THEN3200                       :rem  43
68 IFF$="A"THEN600                        :rem 238
70 IFF$="M"THEN1200                       :rem  32
76 IFF$="{HOME}"THENX=0:Y=0:GOSUB400:GOTO
   45                                     :rem  24
78 IFF$="{CLR}"THENSYS49405:X=160:Y=100:G
   OTO45                                  :rem  83
80 IFF$="S"THEN3000                       :rem  39
82 IFF$="L"THEN3100                       :rem  35
86 IFF$="T"THENPOKECP,21:POKEBM,BO:PRINT"
   {CLR}":GOTO20                          :rem 228
88 IFF$="/"THEN5000                       :rem  13
89 GOTO50                                 :rem  15
90 J2=15-(PEEK(56320)AND15)               :rem 181
95 GETC$:IF C$=""THEN140                  :rem  42
100 IFC$="D"THENMODE$="DRAW":GOTO140
                                          :rem 123
110 IFC$="E"THENMODE$="ERASE":GOTO140
                                          :rem 191
130 IFC$="{F1}"THENN=1-N:GOTO140:rem      254
135 N=1:GOTO50                            :rem  45
140 IFJ2ORN THENJV=J2                     :rem  23
150 X=X+X(JV):Y=Y+Y(JV)                   :rem 149
160 IFMODE$="DRAW"THEN180                 :rem 230
165 GOSUB400                              :rem 176
180 BY=FNB1(0):BI=FNB2(X)                 :rem 172
190 POKEBY,PEEK(BY)OR(2↑BI):GOTO90
                                          :rem 138
300 IFX<0THENX=0:GOTO340                  :rem 228
310 IFX>319THENX=319:GOTO340              :rem 193
320 IFY<0THENY=0:GOTO340                  :rem 232
330 IFY>199THENY=199:GOTO340              :rem 209
335 O=0:GOTO350                           :rem  98
340 O=1                                   :rem  84
350 RETURN                                :rem 120
400 RB=PEEK(BY)-(2↑BI):IFRB<0THENRB=0
                                          :rem   8
410 POKEBY,RB:RETURN                      :rem  57
600 POKECP,21:POKEBM,BO:PRINT"{CLR}"
                                          :rem  58
602 PRINTSPC(8)"{RVS}LINEA AD UN ANGOLO":
    PRINT                                 :rem   2
606 X$="":Y$="":INPUT"PUNTO DI PARTENZA:
    (X,Y)";X$,Y$:IFX$=""ANDY$=""THEN620
                                          :rem 141
610 IFX=VAL(X$)ANDY=VAL(Y$)THEN615
                                          :rem  80
612 GOSUB400                              :rem 173
615 X=INT(VAL(X$)*XP):Y=INT(VAL(Y$)*YP)
                                          :rem 127
620 PRINT:INPUT"ANGOLO:";ANG:ANG=ANG*π/18
    0                                     :rem  82
625 GOSUB3360                             :rem 233
630 DX=COS(ANG):DY=-SIN(ANG)*YP/XP
                                          :rem 122
632 BY=FNB1(0):BI=FNB2(X):IFD=1THENPOKEBY
    ,PEEK(BY)OR(2↑BI):GOTO635   :rem      125
633 GOSUB400                              :rem 176
635 GETS$:IFS$<>""THEN50                  :rem 135
638 GOSUB300:IFOUT=1THEN50                :rem 119
640 X=X+DX:Y=Y+DY:GOTO632                 :rem  77
800 POKECP,21:POKEBM,BO:PRINT"{CLR}"
                                          :rem  60
802 PRINTSPC(13)"{RVS}PARABOLA":PRINT
                                          :rem  33
805 PRINT"EQUAZ.PARABOLA: X=A*Y↑2; DEFINI
    SCI 'A'"                              :rem 223
810 INPUT A:IFA=0THEN PRINT"USA 'O' O 'A'
     PER PLOTTARE UNA LINEA":GOTO810
                                          :rem 240
818 SG=SGN(A):PRINT                       :rem 185
820 INPUT"VERTICE: (X,Y)";CX,CY:CX=CX*XP:
    CY=CY*YP:PRINT                        :rem  78
830 PRINT"SPECIFICA ANGOLO TRA ASSE DI SI
    MMETRIA{ 2 SPAZI}E ASSE ORIZZONTALE:"
                                          :rem  10
835 INPUTANG:ANG=ANG*π/180                :rem  26
838 GOSUB3360                             :rem 239
840 GOSUB400:X=0                          :rem 175
845 Y=SQR(ABS(X/A)):GOSUB4450:IFO=1THENO1
    =1                                    :rem 151
848 Y=-Y:GOSUB4450:IFO=1THENO2=1:rem      235
850 X=X+1*SG/XP                           :rem 129
852 IFO1+O2=2THENO1=0:O2=0:X=CX:Y=CY:GOTO
    45                                    :rem 191
855 GETS$:IFS$<>""THENX=CX:Y=CY:GOTO45
                                          :rem 158
860 GOTO845                               :rem 120
900 POKECP,21:POKEBM,BO:PRINT"{CLR}"
                                          :rem  61
902 PRINTSPC(12)"{RVS}HPERBOLE":PRINT
                                          :rem  48
905 PRINT"EQUAZ.IPERBOLE: 'X↑2/A↑2-Y↑2/B↑
    2=1';{ 4 SPAZI}SPECIFICA 'A','B' (A,B
    )"                                    :rem 119
910 INPUTA,B:IFA=0ORB=0THENPRINT"A,B<>0":
    GOTO910                               :rem   6
920 PRINT:PRINT"ANGOLO TRA ASSE TRASVERSA
    LE E ASSE ORIZ-ZONTALE:"    :rem      139
```

```
930 INPUTANG:ANG=ANG*π/180          :rem   22
940 PRINT:INPUT"COORDINATE CENTRO (X,Y) "
    ;CX,CY:CX=CX*XP:CY=CY*YP        :rem  184
945 GOSUB3360:BY=FNB1(0):BI=FNB2(X):GOSUB
    400                             :rem  137
948 X=SQR(A↑2+B↑2):Y=0:GOSUB4450:X=-X:GOS
    UB4450                          :rem  177
950 X=A                             :rem  116
960 Y=SQR((X↑2/A↑2-1)*B↑2):GOSUB4450:IFO=
    1THENO1=1                       :rem   57
962 Y=-Y:GOSUB4450:IFO=1THENO2=1:rem  232
970 X=-X:GOSUB4450:IFO=1THENO3=1:rem  230
972 Y=-Y:GOSUB4450:IFO=1THENO4=1:rem  235
975 IFO1+O2+O3+O4=4THENO1=0:O2=0:O3=0:O4=
    0:X=CX:Y=CY:GOTO45              :rem  117
980 GETS$:IFS$<>""THENX=CX:Y=CY:GOTO45
                                    :rem  157
990 X=-X:X=X+1/XP:GOTO960           :rem   40
1000 POKECP,21:POKEBM,BO:PRINT"{CLR}"
                                    :rem  101
1005 PRINTSPC(8)"{RVS}LINEA TRA DUE PUNTI
     ":PRINT                        :rem  157
1010 X$="":Y$="":INPUT"PUNTO INIZIALE (X,
     Y) ";X$,Y$:PRINT               :rem   69
1012 IFX$=""ANDY$=""THENX1=X:Y1=Y:GOTO102
     5                              :rem    3
1015 IFX=VAL(X$)ANDY=VAL(Y$)THEN1020
                                    :rem  167
1016 GOSUB400                       :rem  220
1020 X1=INT(VAL(X$)*XP):Y1=INT(VAL(Y$)*YP
     ):X=X1:Y=Y1                    :rem  186
1025 INPUT"PUNTO FINALE (X,Y): ";X2,Y2:X2
     =INT(X2*XP):Y2=INT(Y2*YP)      :rem  250
1100 IFX2-X1=0THENDX=0:DY=SGN(Y2-Y1)*1:GO
     TO1120                         :rem   87
1106 SLP=(Y2-Y)/(X2-X):SY=SGN(Y2-Y):SX=SG
     N(X2-X)                        :rem  188
1108 IFABS(SLP)>1THENDY=SY*1:DX=SX*1/ABS(
     SLP):GOTO1120                  :rem  131
1110 DX=SX*1:DY=SY*ABS(SLP)         :rem  162
1120 GOSUB3360                      :rem   16
1130 BY=FNB1(0):BI=FNB2(X):IFD=1THENPOKEB
     Y,PEEK(BY)OR(2↑BI):GOTO1170:rem  210
1140 GOSUB400                       :rem  218
1170 IFINT(X)<>X2THEN1174           :rem  236
1172 IFINT(Y)=Y2THEN45              :rem   79
1173 Y=Y+DY:GOTO1180                :rem  192
1174 IFINT(Y)<>Y2THEN1177           :rem  245
1175 IFINT(X)=X2THEN45              :rem   80
1176 X=X+DX:GOTO1180                :rem  192
1177 X=X+DX:Y=Y+DY                  :rem  117
1180 GETS$:IFS$="{F1}"THEN50        :rem  251
1182 IFS$="{F3}"THENDX=-DX:DY=-DY:X2=X1:Y
     2=Y1:D=1-D                     :rem  251
1190 GOTO1130                       :rem  201
1200 POKECP,21:POKE BM,BO:PRINT"{CLR}"
                                    :rem  103
1210 INPUT"CURSORE IN (X,Y): ";X2,Y2:X2=X
     2*XP:Y2=Y2*YP:SYS 49485        :rem   58
1220 GOSUB400                       :rem  217
1230 X=X2:Y=Y2:GOTO45               :rem   28
1300 POKECP,21:POKEBM,BO:PRINT"{CLR}"
                                    :rem  104
1302 PRINTSPC(9)"{RVS}CERCHIO O (ARCO)":P
     RINT                           :rem  162
1304 PRINT"EQUAZ.CERCHIO:X↑2+Y↑2=R↑2. SPE
     CIFICA IL RAGGIO (R):"         :rem  127
1305 INPUT R:PRINT                  :rem  114
1307 P=1:INPUT"COORDINATE DEL CENTRO: (X,
     Y) ";CX,CY:CX=CX*XP:CY=CY*YP:PRINT
                                    :rem  237
1309 PRINT"ANGOLO POLARE INIZIALE E FINAL
     E IN GRADI(INIZIALE,FINALE)"
                                    :rem  233
1310 A1$="":A2$="":INPUTA1$,A2$:IFA1$=""A
     NDA2$=""THENA1=0:A2=2* :GOTO1360
                                    :rem  172
1315 A1=VAL(A1$):A2=VAL(A2$):IFA1>360ORA2
     >360THEN1310                   :rem  233
1316 IFA2<A1THENA2=A2+360           :rem  145
1318 A1=A1*π/180:A2=A2*π/180:GOTO1360
                                    :rem  106
1320 POKECP,21:POKEBM,BO:PRINT"{CLR}"
                                    :rem  106
1325 A1=0:A2=2* :PRINTSPC(14)"{RVS}ELLISS
     E":PRINT                       :rem  128
1330 PRINT"EQUAZ.ELLISSE: X↑2/A↑2+Y↑2/B↑2
     =1;{ 7 SPAZI}SPECIFICA A,B (A,B):"
                                    :rem  173
1335 INPUTA,B:PRINT                 :rem  210
1336 IFA<BTHENPRINT"IL MINORE NON PUO'ESS
     ERE PIU'GRANDE DEL MAGGIORE!":GOTO13
     35                             :rem  141
1338 IFA=0THENPRINT"L'ASSE MAGGIORE NON P
     UO'ESSERE 0!":PRINT:GOTO1335
                                    :rem   26
1339 P=B/A:R=A                      :rem   25
1340 PRINT"ANGOLO TRA ASSE MAGGIORE ED AS
     SE ORIZZ:":INPUT ANG:ANG=ANG*π/180
                                    :rem  248
1341 PRINT:INPUT"COORDINATE DEL CENTRO: (
     X,Y)";CX,CY:CX=CX*XP:CY=CY*YP:PRINT
                                    :rem  186
1345 PRINT"VUOI PLOTTARE I FUOCHI? (S/N)"
                                    :rem  179
1350 GETD$:IFD$<>"S"ANDD$<>"N"THEN1350
                                    :rem  146
1360 PRINT:PRINT"DENSITA' PUNTI: (TRA 0.1
      E 1) -{ 9 SPAZI}DEFAULT 0.667."
                                    :rem  249
1370 V$="":INPUT V$                 :rem   10
1380 IFV$=""THENV=1.5:GOTO1400      :rem  228
1390 V=VAL(V$):IFV<.1ORV>1THENPRINT"FUORI
      INTERVALLO":GOTO1370          :rem  241
1395 V=1/V                          :rem   27
1400 GOSUB3360                      :rem   17
1405 GOSUB400                       :rem  222
1430 DR=V/R                         :rem  114
1440 FORAG=A1TOA2STEPDR             :rem  207
1445 X1=X:X=R*COS(AG):DI=SGN(X-X1):GOTO14
     50                             :rem  125
1450 Y=SQR(R↑2-X↑2)*(P)*DI          :rem   32
1480 GOSUB4000                      :rem   17
1492 NEXT                           :rem   15
1500 IFF$="C"THEN1600               :rem  121
1520 X=SQR(A↑2-B↑2):Y=0             :rem  116
1530 IFD=0ORD$="S"THENGOSUB4450:X=-X:GOSU
     B4450                          :rem   61
1540 GOTO1600                       :rem  202
1600 X=CX:Y=CY:GOTO45               :rem   63
3000 POKECP,21:POKEBM,BO:PRINT"{CLR}"
                                    :rem  103
3005 SYS49558                       :rem  214
3008 SYS49485                       :rem  216
3080 GOTO50                         :rem  105
3100 POKECP,21:POKEBM,BO:PRINT"{CLR}"
                                    :rem  104
3105 SYS49600                       :rem  203
3106 SYS49485                       :rem  215
3110 GOTO50                         :rem   99
3200 POKECP,21:POKEBM,BO:PRINT"{CLR}"
                                    :rem  105
```

```
3210 PRINTSPC(3)"{RVS}USI LO SCHERMO"SCR;
     :PRINT"{OFF}("XM","YM")":PRINT
                                  :rem 135
3220 PRINT"POSIZ.CURSORE: X="X/XP:PRINTSP
     C(17)"Y="Y/YP                :rem 237
3230 PRINT:PRINT"UN TASTO PER RIENTRARE I
     N BITMAP"                    :rem 117
3240 GETD$:IFD$=""THEN3240        :rem 187
3250 SYS49485:GOTO50              :rem 175
3360 PRINT:PRINT"DRAW O ERASE (D/E)";
                                  :rem 149
3370 GET DIR$:IFDIR$="D"THENMODE$="DRAW":
     D=1:SYS49485:GOTO3390        :rem 221
3380 IFDIR$="E"THENMODE$="ERASE":D=0:SYS
     49485:GOTO3390               :rem 3
3385 GOTO3370                     :rem 217
3390 RETURN                       :rem 175
4000 IFF$="C"THENOX=X:X=X*XP+CX:Y=Y*YP+CY
     :GOTO4500                    :rem 170
4450 XA=(X*COS(ANG)+Y*SIN(ANG))*XP
                                  :rem 19
4460 YA=(-X*SIN(ANG)+Y*COS(ANG))*YP
                                  :rem 67
4470 OX=X:OY=Y:X=XA+CX:Y=YA+CY    :rem 226
4500 GOSUB300:IFO=1THENGOTO4570   :rem 154
4505 BY=FNB1(0):BI=FNB2(X)        :rem 225
4510 IFD=1THENPOKEBY,PEEK(BY)OR(2↑BI):GOT
     O4570                        :rem 145
4550 GOSUB400                     :rem 226
4570 X=OX:Y=OY:RETURN             :rem 158
5000 POKECP,21:POKEBM,BO:PRINT"{CLR}"
                                  :rem 105
5010 PRINT"BATTI 'GOTO 15' PER RIENTRARE
     NEL{ 7 SPAZI}PROGRAMMA":END:rem 33
```

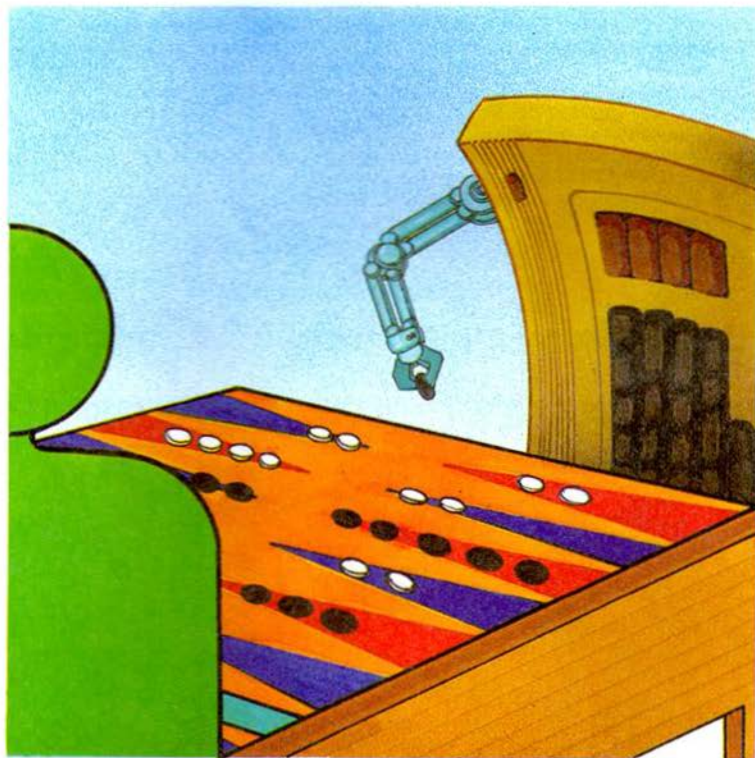




Backgammon per C64, C16 e Plus/4

di **S. Morris**
trad. ed adatt.
di **S. Colombo**

Il gioco del Backgammon ha avuto negli ultimi anni una larghissima diffusione anche in Italia: è un gioco d'azzardo, che può tuttavia essere molto divertente – al contrario del poker – anche se giocato senza rischiare nulla.



Per chi non le conoscesse, ne riassumiamo brevemente le regole. Il cosiddetto "tavoliere" di gioco è suddiviso in quattro diversi settori, due per giocatore.
Rifacendosi a come appare il tavoliere sullo schermo, poiché il computer gioca con le pedine "piene" e voi con quelle "vuote", i vostri due settori sono quelli inferiori: settore interno (lettere X-S) e settore esterno (lettere R-M). Il settore interno del computer è contraddistinto dalle lettere A-F, mentre quello esterno dalle lettere G-L. La configurazione iniziale delle pedine è quella visualizzata. Noterete che si dispone di quindici pedine a testa, ed ogni settore è formato da sei "frecce" di diverso colore. Scopo del gioco è muovere le proprie pedine – a seconda del tiro dei dadi – fino a portarle tutte nel proprio settore interno, quindi – sempre a seconda del tiro dei dadi – toglierle dal tavoliere. Vince chi per primo riesce a togliere dal tavoliere tutte le proprie pedine. Ma procediamo con ordine. Si inizia tirando un dado a testa per stabilire a chi tocca la prima mossa; il tiro dei dadi avviene automaticamente sullo schermo e viene – sempre automaticamente – rieffettuato in caso di parità. Stabilito a chi tocca iniziare, viene visualizzato il tavoliere con la configurazione iniziale. Il senso di movimento delle pedine è mostrato in figura. Da questo momento in poi, per decidere la propria mossa, si tirano sempre due dadi contemporaneamente. Poniamo il caso che abbiate tirato 3 ed 1: potete muovere una pedina di tre "frecce" nel vostro senso di moto ed un'altra pedina (oppure sempre la stessa, ed in questo caso le frecce diventano quattro) di una freccia, sempre nello stesso senso. Non potete andare a "cadere" su frecce occupate da più di una pedina dell'avversario. Esempio: con 3 ed 1 non potete muovere da L in O con il 3 e quindi da O in Q con l'1, perchè Q è occupata da più di una pedina dell'avversario. Potete invece muovere da L in O una pedina, sfruttando il 3, e da A in B un'altra pedina, sfruttando l'1.
Meglio ancora, con 3 ed 1 potete muovere Q in T con il 3 e S in T con l'1, andando così ad occupare la freccia T con due delle vostre pedine, precludendone l'occupazione all'avversario.
Esiste un altro motivo per cui non è consigliabile lasciare pedine isolate su una freccia: se l'avversario capita su una di queste pedine "solitarie", ve la mangia; in questo caso la pedina mangiata viene posta sul "bar", la linea verticale di separazione centrale contraddistinta dalla lettera Y. Al tiro successivo il giocatore mangiato dovrà obbligatoriamente rientrare nel settore interno dell'avversario (e quindi nella posizione più lontana dal proprio settore interno), usando il valore di almeno uno dei due dadi tirati. Esempio: siete stati mangiati e dovete rientrare; avete tirato 6 e 2; con il sei non potete rientrare, poiché la freccia N. 6 è occupata da più di una pedina dell'avversario (ricordiamo che la numerazione delle frecce segue il vo-

stro senso di moto), quindi siete obbligati a rientrare con il 2, e quindi a muovere il 6 con la stessa pedina, oppure con un'altra di vostra scelta.
Una volta che siate riusciti a portare tutte le pedine nel vostro settore interno, con il solito tiro di dadi potete estrarre le pedine dal tavoliere, oppure spostarle – come prima visto – all'interno del settore stesso. Per capire quale pedina è possibile estrarre immaginate di spostarla come sempre nella direzione del moto: se il valore dello spostamento è tale da superare la lettera X, la pedina verrà estratta, altrimenti verrà semplicemente spostata all'interno del settore.
Un'ultima particolarità: nel caso il tiro dei due dadi sia un doppio – ad esempio 4 e 4, o 3 e 3 – esso vale doppio: con 3 e 3 potrete spostare quindi quattro pedine di tre frecce, oppure la stessa pedina (se è possibile) di dodici frecce.
Naturalmente, non è possibile spiegare qui le strategie di gioco del Backgammon, sulle quali esistono decine di volumi: per imparare, come sempre, la cosa migliore è giocare. Non abbiate timore, perché il computer non vi permetterà di compiere mosse illegali, e nel contempo è anche un discrete giocatore.
I giocatori esperti di Backgammon troveranno in questo programma un avversario a volte un po' sprovveduto, ma in grado comunque di divertire ed allenare... Si pensi che l'avversario che abbiamo di fronte non è il solito amico, ma una piccola "intelligenza artificiale" implementata sul nostro computer.
Passiamo ora ad esaminare i semplici comandi del programma. Per tirare i dadi è sufficiente premere la barra spaziatrice, e per muovere la pedina basta premere il tasto corrispondente alla lettera della freccia sulla quale si trova la pedina stessa. Poiché il calcolatore vi chiede sempre quale sia la mossa con il primo dado tirato, se desiderate muovere prima con il valore del secondo dado, premete Z. Se premete Z due volte, significa che siete bloccati e non potete muovere alcuna pedina; in questo caso la mossa passerà al computer.
Infine, se una delle vostre pedine è sul bar e dovete rientrare, scegliete – eventualmente – tramite il tasto Z con quale tiro rientrare e premete Y: la pedina sul bar rientrerà in base al tiro visualizzato sullo schermo.
Questo è uno dei programmi di Backgammon scritti in BASIC più veloci che ci sia capitato di vedere, e l'attesa per la mossa del computer è minima, rendendo rapida ed appassionante ogni partita giocata. Se vi appassionate al gioco e desiderate saperne di più sulle sue strategie, esistono in commercio molti libri sull'argomento, che coprono ogni possibile eventualità.

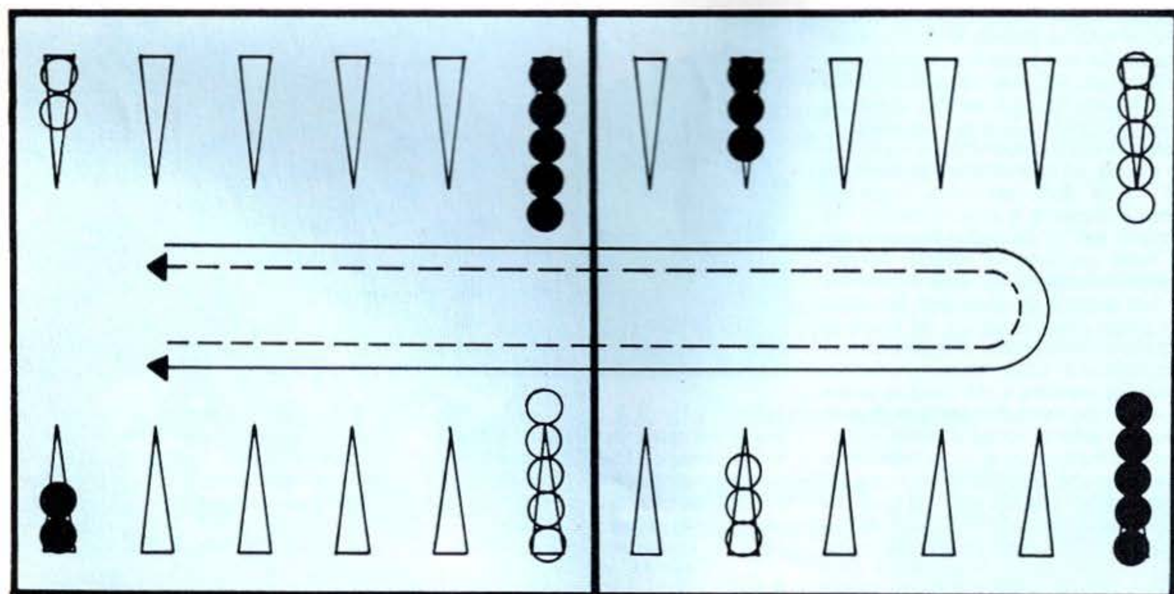
Note per l'utilizzo su C16-PLUS/4

Il programma, con una piccolissima modifica, gira perfettamente anche su C16 e Plus/4. Chi non disponesse della cassetta allegata alla rivista – su quest'ultima è già presente il programma modificato insieme agli altri programmi per C16-Plus/4 – dovrà semplicemente modificare la linea 100 del programma nel modo seguente:

```
100 COLOR0,1 : COLOR4,1 : COLOR1, 2 : WZ = 1
```

ed eliminare la linea 320. Con queste modifiche, il programma sarà reso perfettamente compatibile con questi due calcolatori. Buon divertimento!

configurazione iniziale del Backgammon



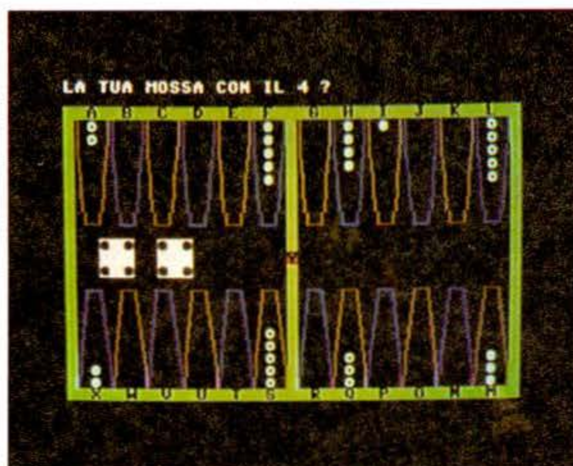
———— senso del moto del giocatore
- - - - - - - - -senso del moto del computer

Backgammon per C64, C16 e Plus/4

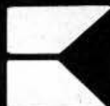
```
100 POKE53280,0:POKE53281,0:POKE646,1:WZ=
    1                                 :rem  165
110 W(1)=-300:W(2)=500:W(3)=100 :rem   49
120 W(4)=-300:W(5)=-500:W(6)=-1200
                                  :rem  199
130 W(7)=70:W(8)=700              :rem    5
140 DIMA$(6),A%(25),P%(12),Q%(12)
                                  :rem   80
150 U$="{ 7 SU}":D$="{ 12 GIU'}":rem  118
160 DEFFNA(X)=INT(X*RND(7)+1)     :rem  167
170 A$(1)="{RVS}{ 3 SPAZI}{GIU'}{ 3 SIN}
    Q {GIU'}{ 3 SIN}{ 3 SPAZI}":rem  179
180 A$(2)="{RVS}{ 2 SPAZI}Q{GIU'}
    { 3 SIN}{ 3 SPAZI}{GIU'}{ 3 SIN}Q
    { 2 SPAZI}"                   :rem  134
190 A$(3)="{RVS}Q{ 2 SPAZI}{GIU'}
    { 3 SIN} Q {GIU'}{ 3 SIN}{ 2 SPAZI}Q
    "                             :rem   89
200 A$(4)="{RVS}Q Q{GIU'}{ 3 SIN}
    { 3 SPAZI}{GIU'}{ 3 SIN}Q Q":rem   35
210 A$(5)="{RVS}Q Q{GIU'}{ 3 SIN} Q
    {GIU'}{ 3 SIN}Q Q"            :rem  246
220 A$(6)="{RVS}Q Q{GIU'}{ 3 SIN}Q Q
    {GIU'}{ 3 SIN}Q Q"            :rem  201
230 Z$="{ 10 SPAZI}":GOTO2450     :rem  210
240 REM                           :rem  122
250 F=0:BH=0:BP=0:NH=15:NP=15     :rem  162
260 RESTORE:FORI=1TO12:READA%(I):rem  240
270 A%(25-I)=-A%(I):NEXT          :rem   16
280 DATA2,0,0,0,0,-5,0,-3,0,0,0,5
                                  :rem   65
290 PRINT"{GIU'}SPAZIO PER INIZIARE ";
                                  :rem  208
300 PRINT"LA PARTITA..."          :rem  144
310 A$="":GETA$:IFA$<>" "THEN310:rem  168
320 POKE53272,21                  :rem   86
330 PRINT"{CLR}{ 8 GIU'}{ 3 SPAZI}C64
    { 3 SPAZI}TU"                 :rem  216
340 PRINT"{HOME}IO GIOCO CON I {CYN}Q
    {WHT}, TU CON I {CYN}W{WHT}":rem  155
350 GOSUB1730:IFP=QTHEN350        :rem   81
360 GOSUB2140:GOSUB2140:GOSUB2160
                                  :rem  228
370 GOSUB1910:F=1:FM=1:IFP>QTHEN930
                                  :rem  129
380 GOTO430                       :rem  107
390 REM MOSSA DEL GIOCATORE       :rem  117
400 GOSUB2120                     :rem  217
410 PRINT"{WHT}SPAZIO PER TIRARE I DADI"
                                  :rem   74
420 A$="":GETA$:IFA$<>" "THEN420:rem  172
430 GOSUB1730:K=0:M=2:IFP=QTHENM=4
                                  :rem   94
440 FM=0:GOSUB2120:GOSUB900       :rem  106
450 PRINT"{WHT}LA TUA MOSSA CON IL";P;"?
    {CYN}";                       :rem    1
460 A$="":GETA$:IFA$=""THEN460    :rem  119
470 N=ASC(A$)-64                  :rem   74
480 IFN>26ORN<1THEN440            :rem  114
490 PRINTA$:IFN=25THEN720         :rem   18
500 IFN=26THEN760                 :rem  227
510 IFBH=0THEN540                 :rem  228
520 GOSUB2120:PRINT"{WHT}SEI SUL BAR!
    {CYN}"                        :rem   86
530 GOSUB2140:GOTO440             :rem  234
540 REM                           :rem  125
550 IFA%(N)>0THEN580              :rem  104
560 GOSUB2120:PRINT"{WHT}NULLA QUI!!{CYN}
    "                             :rem   60
570 GOTO530                       :rem  109
580 IFN+P<25THEN620               :rem   95
590 IFHI=NHTHENF=0:GOTO800        :rem  126
600 GOSUB2120:PRINT"{WHT}FUORI DAL TAVOLO
    {CYN}"                        :rem  181
610 GOTO530                       :rem  104
620 IFA%(N+P)>-2THEN650           :rem   14
630 GOSUB2120:PRINT"{WHT}SPIACENTE,E' BLO
    CCATO!{CYN}"                  :rem   41
640 GOTO530                       :rem  107
650 A%(N)=A%(N)-1:GOSUB1910:N=N+P
                                  :rem   83
660 IFA%(N)>-1THEN690             :rem  154
670 A%(N)=0:BP=BP+1:GOSUB2120     :rem  133
680 PRINT"{WHT}MI HAI MANGIATO!{CYN}"
                                  :rem  236
690 A%(N)=A%(N)+1:M=M-1:GOSUB1910
                                  :rem   54
700 P=Q:IFM>0THEN440              :rem  192
710 GOTO930                       :rem  109
720 IFBHTHEN740                   :rem  124
730 GOTO560                       :rem  110
740 IFA%(P)<-1THEN630             :rem  147
750 N=0:BH=BH-1:GOTO650           :rem   78
760 IFK=0THEN790                  :rem  179
770 GOSUB2120:PRINT"{WHT}PERDI LA MOSSA!
    {CYN}"                        :rem   55
780 GOSUB2140:GOTO930             :rem  245
790 K=P:P=Q:Q=K:GOTO440           :rem  174
800 IFN+P>25THEN850               :rem   97
810 NH=NH-1:IFNHTHEN840           :rem  138
820 GOSUB2120:PRINT"{WHT}HAI VINTO!{CYN}"
                                  :rem   17
830 GOSUB2140:GOTO2340            :rem   30
840 A%(N)=A%(N)-1:N=0:GOTO690     :rem   72
850 FORI=19TON-1                  :rem  195
860 IFA%(I)<=0THEN890             :rem  166
870 GOSUB2120:PRINT"{WHT}NON PUOI.PUOI
    {CYN}";                       :rem   97
880 PRINT"{WHT}MUOVERE IN {CYN}";CHR$(I+6
    4):GOTO530                    :rem   69
890 NEXT:GOTO810                  :rem  236
900 HI=0:FORI=19TO24              :rem  177
910 IFA%(I)>0THENHI=HI+A%(I)      :rem   80
920 NEXT:RETURN                   :rem  244
930 REM                           :rem  128
940 K=0:GOSUB2120:PRINT"{WHT}E' IL MIO TU
    RNO{CYN}"                     :rem    1
950 GOSUB1730:M=2:IFP=QTHENM=4    :rem  115
960 FM=0:IFBP>0THEN1170           :rem   96
970 PI=0:FORI=1TO6                :rem   87
980 IFA%(I)<0THENPI=PI-A%(I)      :rem  103
990 NEXT:IFPI=NPTHENF=0:GOTO1250:rem   59
1000 GOSUB1810:GOSUB1390          :rem  146
1010 IFQM>0THEN1140               :rem   82
1020 IFPM>0THENN=P%(X):GOTO1050 :rem  110
1030 GOSUB2120:PRINT"{WHT}NON POSSO MUOVE
     RE{CYN}"                     :rem   90
1040 GOSUB2140:GOTO390            :rem   27
1050 GOSUB2120:PRINT"{WHT}CON IL"P;"{CYN}
     ";                           :rem   57
1060 PRINT"{WHT}MUOVERO' ";CHR$(64+N)"
     {CYN}"                       :rem   68
1070 A%(N)=A%(N)+1:GOSUB1910:N=N-P
                                  :rem  128
1080 IFA%(N)<1THEN1110            :rem  188
1090 A%(N)=0:BH=BH+1:GOSUB2120    :rem  162
1100 PRINT"{WHT}MANGIATO!{CYN}"  :rem  168
1110 A%(N)=A%(N)-1:M=M-1:GOSUB1910
                                  :rem   92
1120 K=1:P=Q:IFM>0THEN960         :rem  231
```

```
1130 GOTO390                              :rem  154
1140 IFPM=0THEN1160                       :rem   86
1150 IFPM<QMTHENN=P%(X):GOTO1050:rem  222
1160 N=Q%(Y):K=P:P=Q:Q=K:GOTO1050
                                          :rem  233
1170 IFA%(25-Q)<2THEN1210                 :rem   85
1180 IFA%(25-P)<2THEN1200                 :rem   84
1190 GOTO1030                             :rem  200
1200 N=25:BP=BP-1:GOTO1050                :rem  231
1210 IFA%(25-P)<2THEN1230                 :rem   81
1220 K=P:P=Q:Q=K:GOTO1200                 :rem  254
1230 GOSUB1810:IFQM<PMTHEN1220            :rem   68
1240 GOTO1200                             :rem  195
1250 IFA%(P)>=0THEN1320                   :rem  254
1260 GOSUB2120:PRINT"{WHT}USCIRO'{CYN}";
                                          :rem  244
1270 PRINT"{WHT}CON IL"P"{CYN}"           :rem   72
1280 NP=NP-1:IFNPTHEN1310                 :rem  253
1290 GOSUB2120:PRINT"{WHT}HO VINTO!{CYN}"
                                          :rem    8
1300 GOSUB2140:GOTO2410                   :rem   69
1310 N=P:A%(N)=A%(N)+2:GOTO1110           :rem  180
1320 FORI=6TO1STEP-1                      :rem  215
1330 IFA%(I)<0ANDI>PTHEN1000              :rem   92
1340 IFA%(I)<0THEN1360                    :rem  188
1350 NEXT                                 :rem    8
1360 GOSUB2120:PRINT"{WHT}FARO' USCIRE
     {CYN}";                              :rem   19
1370 PRINTCHR$(64+I);" {WHT}CON IL"P"
     {CYN}"                               :rem  180
1380 P=I:GOTO1280                         :rem  224
1390 REM                                  :rem  177
1400 IFKTHEN1420                          :rem  149
1410 GOSUB2120:PRINT"{WHT}STO PENSANDO...
     {CYN}"                               :rem  146
1420 IFPM=0THEN1560                       :rem   91
1430 IFHI=NHTHEN1710                      :rem  179
1440 E=-9999:FORI=1TOPM:H=0               :rem   99
1450 U=P%(I):V=W(7)*U                     :rem  150
1460 A%(U)=A%(U)+1:IFA%(U-P)<1THEN1480
                                          :rem   55
1470 H=1:A%(U-P)=0:V=V+W(8):IFU>18+PTHENV
     =V+450                               :rem  252
1480 A%(U-P)=A%(U-P)-1                    :rem  122
1490 FORJ=1TO24:IFA%(J)>=0THEN1520
                                          :rem  226
1500 L=6:IFA%(J)>-6THENL=-A%(J)           :rem  214
1510 V=V+W(L)-200*(L=1)*(J<7)             :rem  251
1520 NEXTJ:IFV>ETHENE=V:X=I               :rem   18
1530 A%(U)=A%(U)-1                        :rem  124
1540 IFH=1THENA%(U-P)=0                   :rem   52
1550 A%(U-P)=A%(U-P)+1:NEXTI              :rem   56
1560 IFKOR(QM=0)THEN1700                  :rem  154
1570 D=-9999:FORI=1TOQM:H=0               :rem  103
1580 U=Q%(I):V=W(7)*U                     :rem  155
1590 A%(U)=A%(U)+1:IFA%(U-Q)<1THEN1610
                                          :rem   55
1600 H=1:A%(U-Q)=0:V=V+W(8):IFU>18+QTHENV
     =V+450                               :rem  249
1610 A%(U-Q)=A%(U-Q)-1                    :rem  119
1620 FORJ=1TO24:IFA%(J)>=0THEN1650
                                          :rem  225
1630 L=6:IFA%(J)>-6THENL=-A%(J)           :rem  218
1640 V=V+W(L)-200*(L=1)*(J<7)             :rem  255
1650 NEXTJ:IFV>DTHEND=V:Y=I               :rem   21
1660 A%(U)=A%(U)-1                        :rem  128
1670 IFH=1THENA%(U-Q)=0                   :rem   57
1680 A%(U-Q)=A%(U-Q)+1:NEXTI              :rem   62
1690 QM=12:IFD>ETHENPM=12                 :rem   10
1700 RETURN                               :rem  168
```

```
1710 X=PM:Y=QM:PM=SGN(PM):QM=SGN(QM)
                                          :rem   63
1720 GOTO1700                             :rem  203
1730 IFFMTHEN1780                         :rem  236
1740 P=FNA(6):Q=FNA(6):T=TI               :rem   65
1750 IFTI-T>120THEN1770                   :rem   73
1760 P=P+1:IFP>6THENP=1                   :rem  129
1770 Q=Q+1:IFQ>6THENQ=1                   :rem  134
1780 PRINT"{HOME}{ 3 DES}";D$;"{WHT}"A$(P
     );"{ 2 DES}{ 2 SU}";A$(Q)            :rem   85
1790 IFTI-T<410THEN1750                   :rem   75
1800 RETURN                               :rem  169
1810 REM                                  :rem  174
1820 PM=0:QM=0:FORI=1TO24                 :rem  251
1830 IFA%(I)>=0THEN1900                   :rem  255
1840 IFI-P<1THEN1870                      :rem  142
1850 IFA%(I-P)>1THEN1870                  :rem   72
1860 PM=PM+1:P%(PM)=I                     :rem  197
1870 IFK>0ORI-Q<1THEN1900                 :rem  230
1880 IFA%(I-Q)>1THEN1900                  :rem   70
1890 QM=QM+1:Q%(QM)=I                     :rem  204
1900 NEXT:RETURN                          :rem   35
1910 REM                                  :rem  175
1920 B$="{HOME}{ 3 GIU'}":W$="{GIU'}{SIN}
     ":T=0:GOSUB2020                      :rem   86
1930 B$=B$+D$+LEFT$(D$,8):W$="{SU}{SIN}"
                                          :rem  100
1940 T=1:GOSUB2020                        :rem   14
1950 PRINT"{HOME}";D$;SPC(21);            :rem  102
1960 IFBP=0THENPRINTZ$:GOTO1980           :rem  221
1970 FORI=1TOBP:PRINT"{CYN}Q {SIN}";:NEXT
                                          :rem  112
1980 PRINT"{HOME}";D$;"{ 2 GIU'}";SPC(21)
     ;                                    :rem   10
1990 IFBH=0THENPRINTZ$:GOTO2010           :rem  201
2000 FORI=1TOBH:PRINT"{CYN}W {SIN}";:NEXT
                                          :rem   95
2010 RETURN                               :rem  163
2020 FORI=1TO12:V=T*(25-I)-I*(T=0)
                                          :rem  147
2030 IFV<>NANDF>0THEN2110                 :rem  236
2040 A$=" "                               :rem  172
2050 PRINTB$SPC(3*I-(I>6)-1);             :rem  107
2060 IFA%(V)=0THEN2100                    :rem  195
2070 K=209:IFA%(V)>0THENK=215             :rem  127
2080 FORJ=1TOABS(A%(V))                   :rem   64
```



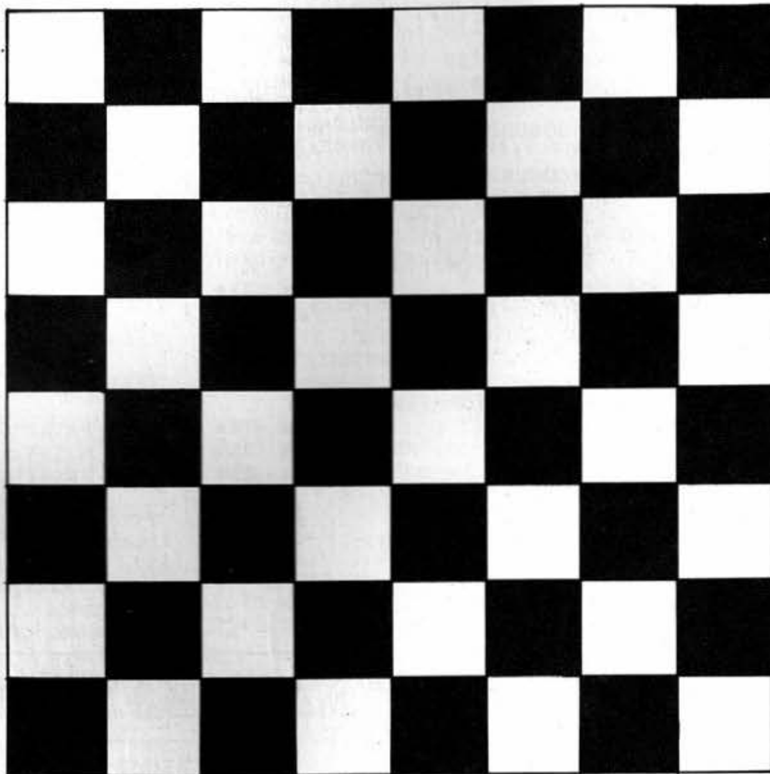
```
2090 PRINT"{CYN}"CHR$(K);W$;:NEXT
                                  :rem  66
2100 PRINTA$                      :rem 181
2110 NEXT:RETURN                  :rem  29
2120 REM                          :rem 169
2130 PRINT"{HOME}"Z;Z$;Z$;Z$"{HOME}";:RET
     URN                          :rem 219
2140 FORI=1TO2000:NEXT            :rem  67
2150 RETURN                       :rem 168
2160 REM                          :rem 173
2170 W$="T [<M>]{GIU'}{ 3 SIN}T [<M>]
     {GIU'}{ 3 SIN}T Y{GIU'}{ 3 SIN}G H
     {GIU'}{ 3 SIN}B -{GIU'}{ 3 SIN}- B
     {GIU'}{ 3 SIN}H G{GIU'}{ 3 SIN}[<M>]
     [<@>][<G>]"                  :rem  13
2180 PRINT"{CLR}{ 2 GIU'}":FORI=1TO21
                                  :rem  47
2190 PRINT"{GRN}{RVS} ";SPC(18);" ";SPC(1
     8);" "                       :rem 129
2200 NEXT:PRINT"{HOME}{GIU'}":J=65:K=1
                                  :rem  81
2210 GOSUB2280:GOSUB2280          :rem 151
2220 PRINT" ";D$;LEFT$(D$,9):J=88:K=-1
                                  :rem  51
2230 GOSUB2280:GOSUB2280:PRINT" {HOME}
     { 2 GIU'}"                   :rem 217
2240 GOSUB2300:GOSUB2300:PRINTD$:rem 187
2250 W$="[<M>][<T>][<G>]{GIU'}{ 3 SIN}Y
     T{GIU'}{ 3 SIN}H G{GIU'}{ 3 SIN}B -
     {GIU'}{ 3 SIN}G H{GIU'}{ 3 SIN}T Y
     {GIU'}{ 3 SIN}T [<M>]{GIU'}{ 3 SIN}
     [<G>] [<M>]"                 :rem 234
2260 GOSUB2300:GOSUB2300          :rem 142
2270 PRINT"{HOME}{GIU'}";D$;SPC(19);"
     {RVS}{RED}Y{OFF}{CYN}":RETURN
                                  :rem 144
2280 FORI=1TO6:PRINT"{GRN}{RVS}
     { 2 SPAZI}";CHR$(J);         :rem 144
2290 J=J+K:NEXT:PRINT" ";:RETURN:rem 237
2300 PRINT"{DES}";:FORI=1TO3      :rem 156
2310 IFWZ<3THENPRINT"[<1>]"W$;U$;"{BLU}"W
     $;U$;                        :rem  45
2320 IFWZ>=3THENPRINT"{BLU}"W$;U$;"[<1>]"
     W$;U$;                       :rem 109
2330 NEXT:WZ=WZ+1:RETURN          :rem  86
2340 REM                          :rem 173
2350 PRINT"{CLR}GRAZIE PER LA PARTITA!
     {GIU'}                       :rem 137
2360 PRINT"SEI UN BUON AVVERSARIO!"
                                  :rem 119
2370 PRINT"GIOCHI ANCORA? (S O N)
                                  :rem  98
2380 A$="":GETA$:IFA$=""THEN2380:rem 221
2390 IFA$="S"THENRUN              :rem 186
2400 END                          :rem 157
2410 REM                          :rem 171
2420 PRINT"{CLR}ACCIDENTI! HO VINTO!!
     {GIU'}"                      :rem  75
2430 PRINT"MI SONO PROPRIO DIVERTITO!"
                                  :rem 117
2440 GOTO2370                     :rem 207
2450 REM                          :rem 175
2460 PRINT"{CLR}";TAB(13);        :rem  50
2470 PRINT"{RVS}{RED}BACKGAMMON{WHT}
     {GIU'}"                      :rem 178
2480 PRINT"IO GIOCO CON LE REGOLE STANDAR
     D CHE DEVI"                  :rem  32
2490 PRINT"CONOSCERE PRIMA DI INIZIARE A
     GIOCARE.{GIU'}"              :rem  28
2500 PRINT"I TASTI CHE DEVI PREMERE SONO
     QUESTI:{GIU'}"               :rem 211
2510 PRINT" A-X PER MUOVERE LE PEDINE
     {GIU'}"                      :rem 192
2520 PRINT" Y{ 3 SPAZI}PER RIENTRARE DAL
     BAR{GIU'}"                   :rem  61
2530 PRINT" Z{ 3 SPAZI}PER MUOVERE IN ORD
     INE INVERSO{GIU'}"           :rem 142
2540 PRINT" SPAZIO QUANDO RICHIESTO{GIU'}
     "                            :rem 245
2550 PRINT"SE PREMI DUE VOLTE Z SALTI LA
     MOSSA.{GIU'}"                :rem  64
2560 PRINT"CIO' E' CONCESSO SOLO SE SEI B
     LOCCATO.{GIU'}"              :rem 165
2570 PRINT                        :rem  91
2580 GOTO240                      :rem 158
```



DAMA per C16 - Plus/4

di **A. Motta**

Non poteva certamente mancare fra la biblioteca software del C16 un gioco famoso come la dama. Anche se sicuramente la quasi totalità dei lettori conoscerà questo stupendo gioco, è doveroso fare comunque un breve cenno delle regole per quei pochi che non l'hanno mai praticato.
Le origini del gioco vengono fatte risalire all'epoca del Rinascimento. Si pratica su una scacchiera di 64 caselle alternate, una bianca ed una nera, ove si dispongono 24 pedine: 12 nere per un giocatore e 12 bianche per l'avversario.



Ciascun giocatore depone le proprie pedine sulle prime tre file del proprio lato della scacchiera. I movimenti delle pedine debbono avvenire esclusivamente in direzione dello schieramento avversario, spostandosi una alla volta lungo le diagonali delle caselle nere. La pedina che, lasciandosi una casella vuota dietro di sè, viene a trovarsi in una casella attigua a quella di una pedina avversaria, è scavalcata e "mangiata" o "presa" da quest'altra. Le regole prevedono che, obbligatoriamente, le prese devono essere effettuate. Quando una pedina raggiunge una casella vuota della prima fila avversaria, viene sovrapposta ad un'altra e viene chiamata "dama". Questa ha il vantaggio di poter essere mossa sia in avanti sia indietro, di poter "mangiare" pedine e dame avversarie, ma di poter essere "mangiata" solo da un'altra dama.
Vince chi riesce a "mangiare" tutte le pedine dell'avversario o a bloccarle in modo che non sia più possibile muoverle.
Il programma qui presentato implementa fedelmente il gioco della Dama con le regole sopra riportate e vi consente di sfidare il vostro C16.
Il computer non è comunque un eccellente giocatore; il livello di gioco è abbastanza semplice: viene esaminata una determinata situazione della scacchiera ed in base alla stessa viene scelta la mossa senza vagliare i possibili effetti su mosse successive. Questo in quanto una routine che consentisse di migliorare notevolmente il livello di gioco comporterebbe, oltre ad un notevole allungamento del programma con problemi di spazio in memoria (il programma da solo occupa ben 9K a fronte di una grafica che – perdonate la poca modestia – si può definire veramente ben fatta), anche un notevole allungamento dei tempi di risposta essendo il programma fatto totalmente in BASIC.
Comunque il C16 si difende bene e dovranno stare attenti anche i più esperti in quanto basterà una banale svista per perdere partite che si credevano già concluse a proprio favore.

Strategia del C16
Prima di illustrare la strategia che il C16 segue per effettuare una mossa, vorrei consigliare a quei lettori che intendessero utilizzare il programma al solo scopo di gioco, di non proseguire nella lettura dell'articolo in quanto la conoscenza della strategia del computer comporterebbe già in anticipo la determinazione delle mosse in risposta con conseguente facilità nell'impostare il gioco, vincita sicura ed in ultima analisi, mancato divertimento.
Il C16 vede la scacchiera come una matrice di 100 elementi dei quali 64 sono le caselle della stessa ed altre 36 variabili sono il bordo della scacchiera, indispensabile per il corretto funzionamen-

to del gioco in fase di controllo e scelta delle mosse.
Delle 64 caselle della scacchiera, quelle relative allo svolgimento della gara sono unicamente quelle nere che il computer identifica con le variabili della matrice.
All'inizio della gara tutta la matrice viene posta a "9". Quindi le prime tre file nere della scacchiera sono riempite con il valore "0" corrispondente alla pedina bianca, le due file nere centrali hanno il valore "255" – casella vuota, e le restanti tre file nere superiori sono valorizzate con il numero "120" corrispondente alla pedina nera.
Gli altri valori che la matrice può contenere sono "1" per la dama bianca e "129" per la dama nera.
La scelta della mossa da parte del C16 avviene nel seguente modo ed è relativa alle linee del programma da 5000 a 5390.
La routine si compone di due parti. Nella prima (da 5030 a 5180) il C16 prende in rassegna tutta la matrice e controlla se può effettuare una o più prese; nel qual caso, considerato che è obbligatorio mangiare, muove la pedina o la dama in grado di prendere uno o più pezzi avversari. Se quanto sopra non è possibile, passa allora alla seconda parte della routine (dalla riga 5190 alla riga 5390 del programma), nella quale determinerà quale pezzo muovere. Come già esposto in precedenza, il C16 non è in grado di sviluppare una determinata situazione di gioco che consenta di analizzare le conseguenze di una mossa rispetto alle possibili successive, ma si limita a prendere in esame la situazione del momento e valutare, seguendo i criteri più avanti esposti, quale sia la possa migliore da effettuare.
Prima di passare ad analizzare queste linee, occorre fare un cenno sulle direzioni relative alle mosse. Notiamo che la differenza fra due caselle adiacenti diagonalmente ha il valore di 6 o 7 (–6 o –7 se la differenza è eseguita fra caselle basse meno quelle più alte) ed è stata intenzionalmente scelta per facilitare i controlli senza creare confusioni fra caselle di file alterne ai bordi della scacchiera. I suddetti valori sono contenuti nella matrice X(8) e sono stati duplicati per meglio facilitare i controlli della mossa.
Ritornando al discorso iniziale, quando il C16 trova una casella occupata da una pedina propria, controlla che nelle uniche due direzioni possibili di mossa (in basso a sinistra ed in basso a destra), la casella è occupata prosegue con la direzione destra, altrimenti assegna alla casella indicata dalla direzione, un valore strategico iniziale di 12.
A questo punto comincia a controllare tutte le caselle adiacenti a quella occupata dalla pedina in esame ed effettua le seguenti considerazioni.
– se la casella, successiva a quella libera indicata dalla direzione, è occupata da una pedina o da una dama avversaria, il valore stratetigico viene diminuito di 2 (due);
– se la casella adiacente orizzontalmente è occupata da una dama avversaria a quella sottostante perpendicolarmente è libera, il valore strategico viene diminuito di due;
– se la casella in esame è una dell'ultima fila del lato nero, il valore strategico viene diminuito di uno;
– se la casella sottostante è occupata o da una pedina bianca o da una dama bianca e la casella adiacente orizzontalmente è libera, il valore strategico viene diminuito di due;
– se la casella sottostante è occupata da una pedina bianca e la casella adiacente orizzontalmente è invece occupata, il valore strategico viene aumentato di 0,5;
– se la casella adiacente nella direzione opposta alla direzione che si sta esaminando è occupata da una pedina o dama avversaria e la casella opposta a tale direzione è libera, il valore strategico viene aumentato di uno;
– se la casella indicata dalla direzione è una della prima fila avversaria, in modo che la pedina diventi dama, viene interrotto il ciclo e viene scelta tale direzione come mossa da effettuarsi;
– se la casella adiacente nella direzione opposta alla direzione in esame è occupata da una pedina propria e la successiva in tale direzione è occupata da una pedina o dama avversaria, il valore strategico viene diminuito di tre.
Dopo aver eseguito tutti i controlli sopra riportati, il C16 confronta il valore strategico risultante, contenuto nella variabile LS, con il valore dela variabile MS (che all'inizio della routine di scelta mossa viene portato a zero). Se il valore della variabile LS è inferiore a quello della variabile MS, la direzione viene scartata; se è superiore il C16 considera tale direzione come possibile mossa, ed inserisce nella variabile PM il valore della casella contenente la propria pedina presa in esame, nella variabile PI il valore della casella indicata dalla direzione che si è appena esaminata e nella variabile MS il valore della LS. Se invece il valore di tali variabili è identico, il computer determinerà casualmente quale delle due considerare come possibile mossa.
Nel caso in cui la casella in esame contenga una dama, il C16 effettua una serie di controlli leggermente diversi da quelli sopra esposti.
Questo a seguito di una diversa impostazione nella strategia di gioco: la pedina deve avere a disposizione una maggior difesa rispetto alla dama in quanto può essere mangiata da qualsiasi pezzo avversario a differenza della dama che può essere mangiata solamente da un'altra dama; inoltre deve essere in grado di sviluppare situazioni d'attacco in modo da raggiungere il lato opposto avversario e diventare dama. Quando ciò è avvenuto, subentra un certo tipo di rilassamento e la dama del C16 pensa più a difendersi che ad attaccare.
Ritornando alla routine di scelta mossa, quando la casella in esame è occupata da una dama nera, il C16 effettua i controlli sottoelencati che, in questo caso, sono in quattro direzioni possibili.
In analogia a quanto già evidenziato per la pedina, se la casella adiacente a quella in esame, è libera, viene assegnato un valore strategico di undici. Come avrete potuto notare, tale valore è inferiore di uno rispetto a quello assegnato nel caso in cui la casella in esame contenga una pedina, ed è stato intenzionalmente scelto per poter sviluppare la strategia di gioco illustrata precedentemente.
Dopo aver assegnato il valore strategico, vengono effettuati i seguenti controlli:
– se la casella, successiva a quella libera indicata dalla direzione in esame, è occupata da una dama avversaria, il valore strategico viene diminuito di due;
– se la casella nella direzione opposta a quella indicata dalla direzione in esame, è occupata da una dama avversaria e la dama del C16 ha dietro di sè una casella libera, il valore strategico viene aumentato di uno;
– se la casella sottostante o sovrastante perpendicolarmente è occupata da una dama avversaria e quella adiacente orizzontalmente dal lato della direzione in esame è libera, il valore strategico viene diminuito di due;
– se la casella successiva a quella libera indicata dalla direzione è occupata da una pedina avversaria, il valore strategico viene aumentato di uno;
– se la casella sottostante o sovrastante perpendicolarmente è libera a quella adiacente orizzontalmente dalla parte della direzione in esame è occupata da una dama avversaria, il valore strategico viene diminuito di due;
– se c'è la possibilità di inserirsi fra due dame avversarie (i cosiddetti "occhiali"), il valore strategico viene aumentato di tre.
Eseguiti i suddetti controlli, viene chiamata la subroutine alle linee 5900-5920 per il controllo del valore strategico e scelta della possibile mossa (vedasi quando indicato in precedenza in merito alle variabili LS, MS, PM e PI).
Dopo aver passato in rassegna tutte le caselle della scacchiera, il C16 controlla il valore della variabile MS. Se è zero vuol dire che non è in grado di eseguire alcuna mossa in quanto i pezzi rimasti sono stati bloccati da quelli avversari e, pertanto, decreterà la vostra vittoria. Se invece tale variabile è diversa da zero, il computer leggerà nella variabile PM il numero della casella contenente il pezzo da muovere e, nella variabile PI, il numero della casella di destinazione.

Utilizzo del programma

Anche se all'interno del programma sono contenute brevi istruzioni sul gioco, è bene dare qualche ulteriore spiegazio-

ne sull'utilizzo del programma stesso. All'inizio, dopo la presentazione con le relative istruzioni, vi verrà richiesto se volete iniziare per primi la gara, per cui, se preferite che parta il C16, basterà rispondere "N" (NO) alla domanda.
La mossa si effettua inserendo prima le coordinate della casella contenente il pezzo da muovere e quindi le coordinate della casella di arrivo.
La forma di tali coordinate dovrà essere "Lettera/Numero".
Per non appesantire il programma, non sono stati predisposti i controlli nel caso in cui dobbiate effettuare più prese con un'unica mossa.
Pertanto, se vi capiterà di dover mangiare più di un pezzo del C16 con una sola mossa, dovrete effettuare tante mosse quanti sono i pezzi da prendere. Ad ogni presa che effettuate il C16 vi chiederà se ne avete un'altra in successione e, nel caso sopra riportato, basterà rispondere "S" (SI) alla domanda per inserire le coordinate per effettuare l'altra presa. Idem per prese successive. State ben attenti a non barare: il computer controlla che, nel caso comunichiate una successiva presa, la casella di partenza che dovrete inserire dovrà essere tassativamente uguale a quella di arrivo della presa che avete appena eseguito, e che la differenza fra la caselladi partenza e quella di arrivo sia superiore a sette (cioè vi sia effettivamente una presa). In caso contrario segnalerà il vostro errore. A questo punto basterà premere il tasto "P" (PASSO) al posto delle coordinate in modo che la mossa passi al C16.
Se vi siete stancati di giocare o volete interrompere la partita per iniziarne una nuova, basterà inserire una "X" (EXIT) al posto delle coordinate ed il computer vi chiederà se volete effettuare una nuova gara.
Nel programma sono state inserite diverse linee REM che ne rendono chiara la lettura. Merita comunque un cenno la piccola routine contenuta nelle linee da 4030 a 4150 che simula una istruzione INPUT utilizzando la funzione GET e permette l'inserimento delle coordinate da parte del giocatore.
Come molti avranno già intuito, l'utilizzo della istruzione INPUT avrebbe comportato i seguenti problemi:
1) la presenza dell'antiestetico punto di domanda;
2) la possibilità di immettere errori con conseguente visualizzazione di messaggi del tipo "EXTRA IGNORED" o "REDO FROM START", che si posizionerebbero sopra la scacchiera rovinandola irrimediabilmente, in considerazione della posizione in cui è stata ubicata la visualizzazione delle coordinate inserite;
3) la possibilità di inserire più dei 4 caratteri relativi alle coordinate e, quindi, effettuare delle scritte sopra la scacchiera.
Il primo punto è senz'altro un problema marginale e facilmente risolvibile aprendo un file verso il video e poi, magari non a tutti dava fastidio il punto interrogativo lampeggiante sullo schermo.
Questa soluzione lascia però immutati gli altri problemi, ben più importanti in quanto potrebbero compromettere il gioco stesso.
A fronte di ciò ho creato quella piccola routine che consente di inserire i quattro caratteri richiesti dall'imput delle coordinate algebriche, e permette anche la cancellazione di quelli immessi erroneamente, utilizzando il tasto (HOME/CLEAR). Inoltre, i controlli inseriti consentono l'inserimento per il 1° ed il 3° carattere delle sole lettere dalla A all'H; mentre per il 2° ed il 4° carattere il sistema accetta solamente numeri dall'1 all'8.

Dama per C16 - Plus/4

```
1000 :                                :rem 251
1010 REM TRASFERIMENTO SET CARATTERI
                                      :rem  20
1020 :                                :rem 253
1030 POKE56,60:POKE52,60:POKE55,0:POKE51,
     0                                :rem 172
1040 T=0:FORI=1000TO1030:READA        :rem 236
1050 POKEI,A:T=T+A:NEXT               :rem 175
1060 IFT<>3288THENPRINT"{CLR}ERRORE NEI D
     ATA!":END                        :rem 131
1070 SYS1000:POKE65298,0:POKE65299,60
                                      :rem 101
1090 :                                :rem   4
1100 REM DATA L/M                     :rem 136
1110 :                                :rem 253
1120 DATA160,0,185,0,208,153,0,60
                                      :rem 112
1130 DATA185,0,209,153,0,61,185,0
                                      :rem 122
1140 DATA210,153,0,62,185,0,211,153
                                      :rem 211
1150 DATA0,63,24,136,208,228,96       :rem  39
1190 :                                :rem   5
1200 REM DATA NUOVI CARATTERI         :rem 241
1210 :                                :rem 254
1220 DATA255,255,255,255,255,255,255,255
                                      :rem 243
1230 DATA1,14,48,64,96,88,71,64       :rem  49
1240 DATA255,0,24,36,24,0,0,255       :rem  18
1250 DATA128,112,12,2,6,26,226,2:rem    68
1260 DATA64,64,96,24,6,1,0,0          :rem 135
1270 DATA0,0,0,0,0,255,0,0            :rem   4
1280 DATA2,2,6,24,96,128,0,0          :rem 131
1290 DATA1,15,63,127,95,103,120,127
                                      :rem 227
1300 DATA255,255,235,219,231,255,255,0
                                      :rem 126
1310 DATA128,240,252,254,250,230,30,254
                                      :rem 162
1320 DATA127,127,127,31,7,1,0,0 :rem    14
1330 DATA255,255,255,255,255,255,0,0
                                      :rem  29
1340 DATA254,254,254,248,224,128,0,0
                                      :rem  24
1350 DATA0,0,1,14,48,64,96,88         :rem 194
1360 DATA0,0,255,0,24,36,24,0         :rem 169
1370 DATA0,0,128,112,12,2,6,26        :rem 219
1380 DATA71,64,64,64,32,88,71,64:rem   103
1390 DATA0,255,0,0,0,0,0,255          :rem 115
1400 DATA226,2,2,2,4,26,226,2         :rem 171
1410 DATA0,0,1,15,63,127,95,103 :rem    16
1420 DATA0,0,255,255,231,219,231,255
                                      :rem  17
1430 DATA0,0,128,240,252,254,250,230
                                      :rem   7
1440 DATA120,127,127,127,63,103,120,127
                                      :rem 162
1450 DATA255,0,255,255,255,255,255,0
                                      :rem  32
1460 DATA30,254,254,254,252,230,30,254
                                      :rem 121
1470 FORI=16136TO16335:READA:POKEI,A:NEXT
                                      :rem  14
1500 :                                :rem   0
1510 REM VARIABILI INIZIALI           :rem 151
1520 :                                :rem   2
1530 Z1$="{WHT}[< 3 K>]{GIU'}{ 3 SIN}
     [< 3 K>]{GIU'}{ 3 SIN}[< 3 K>]":Z2$=
     "{WHT}{ 3 SPAZI}{GIU'}{ 3 SIN}[<I>]
     [<T>][<@>]{GIU'}{ 3 SIN}[<G>][<+>]
     [<M>]"                           :rem 146
1540 Z3$="{WHT}{ 3 SPAZI}{GIU'}{ 3 SIN}
     [<£>]£[<N>]{GIU'}{ 3 SIN}[<Q>][<D>]
     [<Z>]":Z4$="{ 3 SPAZI}{GIU'}
     { 3 SIN}{ 3 SPAZI}{GIU'}{ 3 SIN}
     { 3 SPAZI}"                      :rem  13
```

```
1550 Z5$="{WHT}[<S>][<P>][<A>]{GIU'}
     { 3 SIN}[<E>][<R>][<W>]{GIU'}
     { 3 SIN}[<G>][<+>][<M>]":Z6$="{WHT}
     [<H>][<J>][<L>]{GIU'}{ 3 SIN}[<Y>]
     [<U>][<O>]{GIU'}{ 3 SIN}[<Q>][<D>]
     [<Z>]"                          :rem  120
1560 PB=0:PN=128:DB=1:DN=129:B=255
                                     :rem  163
1570 DIMA(100),X(8),B(2),Z(2)        :rem  209
1580 DATA-6,-7,6,7,-7,-6,7,6         :rem  132
1585 RESTORE1580                     :rem  197
1590 FORI=1TO8:READX(I):NEXT         :rem    9
2000 :                               :rem  252
2010 REM PRESENTAZIONE               :rem  142
2020 :                               :rem  254
2030 COLOR4,7,2:COLOR0,7,2:VOL8      :rem   70
2040 PRINT"{CLR}":CHAR,17,3,"{WHT}D A M A
     "                               :rem  253
2050 CHAR,17,4,"{ 7 *}":PRINT        :rem   80
2060 PRINT"{YEL}{ 3 GIU'}TU GIOCHI CON LE
      PEDINE BIANCHE E IL C16";      :rem  149
2070 PRINT"CON QUELLE NERE."         :rem  154
2080 PRINT"PER EFFETTUARE LA MOSSA
     { 2 SPAZI}PRIMA INSERISCI";:rem  218
2090 PRINT"LE COORDINATE DELLA CASELLA
     { 2 SPAZI}DI PARTENZA";         :rem  153
2100 PRINT"E POI QUELLE DELLA CASELLA DI
     ARRIVO."                        :rem  110
2105 PRINT"PER CANCELLARE USA IL TASTO <C
     LEAR/HOME>";                    :rem  199
2110 PRINT"EFFETTUARE UNA PRESA PER VOLTA
     :{ 2 SPAZI}IL{ 2 SPAZI}C16";
                                     :rem    0
2120 PRINT"VI CHIEDERA' SE NE AVETE UNA
     { 2 SPAZI}SUCCESSIVA";          :rem   73
2130 PRINT"DA EFFETTUARE."           :rem   53
2140 PRINT"LE PRESE DEVONO ESSERE ESEGUIT
     E OBBLIGA-";                    :rem  237
2150 PRINT"TORIAMENTE."              :rem  191
2160 PRINT"{ 3 GIU'}{CYN}PREMI <RETURN> P
     ER INIZIARE"                    :rem  133
2170 GETZ$:IFZ$<>CHR$(13)THEN2170
                                     :rem  152
2180 PRINT"{CLR}":CHAR,14,10,"{FLASH ON}
     {YEL}A T T E N D I "            :rem  194
2500 :                               :rem    1
2510 REM VARIABILI GIOCO             :rem  176
2520 :                               :rem    3
2530 FORI=1TO100:A(I)=9              :rem   39
2540 IFI<73ANDI>55ANDI<>68ANDI<>67ANDI<>6
     2ANDI<>61ANDI<>60THENA(I)=PN:GOTO257
     0                               :rem   49
2550 IFI<54ANDI>42ANDI<>49ANDI<>48ANDI<>4
     7THENA(I)=B:GOTO2570            :rem  217
2560 IFI<41ANDI>23ANDI<>36ANDI<>35ANDI<>3
     4ANDI<>29ANDI<>28THENA(I)=PB
                                     :rem  217
2570 NEXT                            :rem   13
2580 HS=0:CS=0                       :rem   20
3000 :                               :rem  253
3010 REM STAMPA SCACCHIERA           :rem   52
3020 :                               :rem  255
3030 COLOR0,1:COLOR4,4,6:PRINT"{CLR}"
                                     :rem   31
3040 FORI=0TO23:CHAR,0,I,"{CYN}[<K>]":CHA
     R,25,I,"[< 15 K>]":NEXT         :rem  253
3050 FORI=0TO18STEP6:FORJ=1TO20STEP6
                                     :rem   51
3060 CHAR,J,I,Z1$:NEXTJ,I            :rem  229
3070 FORI=3TO22STEP6:FORJ=4TO23STEP6
                                     :rem   57
3080 CHAR,J,I,Z1$:NEXTJ,I            :rem  231
3090 CHAR,0,24,"{RVS}{CYN}{ 2 SPAZI}A
     { 2 SPAZI}B{ 2 SPAZI}C{ 2 SPAZI}D
     { 2 SPAZI}E{ 2 SPAZI}F{ 2 SPAZI}G
     { 2 SPAZI}H{ 15 SPAZI}{OFF}"
                                     :rem  175
3100 POKE3047,99:POKE4071,97         :rem   48
3110 PRINT"{RVS}{HOME}{GIU'}8{ 3 GIU'}
     {SIN}7{ 3 GIU'}{SIN}6{ 3 GIU'}{SIN}5
     { 3 GIU'}{SIN}4{ 3 GIU'}{SIN}3
     { 3 GIU'}{SIN}2{ 3 GIU'}{SIN}1{OFF}"
                                     :rem  178
3120 CHAR,29,1,"{RVS}D A M A":CHAR,29,2,"
     { 7 E}{OFF}"                    :rem   31
3130 CHAR,31,5,"{RVS}C16":CHAR,31,12,"T U
     {OFF}"                          :rem   36
3140 GOSUB8100                       :rem   17
3150 CHAR,28,19,"{RVS}{CYN}PARTI PER":CHA
     R,26,20,"PRIMO?{ 2 SPAZI}(S/N){OFF}"
                                     :rem  254
3160 SOUND1,810,15:SOUND1,770,15:rem   43
3170 GETZ$:IFZ$<>"S"ANDZ$<>"N"THEN3170
                                     :rem  216
3180 CHAR,28,19,"[< 9 K>]":CHAR,26,20,"
     [< 13 K>]"                      :rem   70
3190 IFZ$="S"THEN4000                :rem  161
3200 I=56:GOTO5195                   :rem  253
3990 :                               :rem   15
4000 REM MOSSA GIOCATORE             :rem  200
4010 :                               :rem  255
4020 MO=0:CHAR,30,14,"{CYN}[< 5 K>]{SIN}
     {FLASH ON}{ 4 SIN}{RVS}<{RVS}":SOUND
     1,800,25                        :rem   43
4030 GETZ$:IFZ$="X"THEN7560          :rem   70
4035 IFZ$="P"ANDR=2THENCHAR,30,14,"
     { 2 SPAZI}":GOTO5000            :rem   83
4050 IFZ$<"A"ORZ$>"H"THEN4030        :rem  118
4060 M$=Z$+"{FLASH ON}<{FLASH ON}":CHAR,3
     0,14,M$                         :rem  188
4070 GETZ$:IFZ$="{HOME}"THENZ$="":CHAR,30
     ,14,"{FLASH ON}<{FLASH ON} ":GOTO403
     0                               :rem  150
4080 IFZ$<"1"ORZ$>"8"THEN4070        :rem   93
4090 M$=LEFT$(M$,1)+Z$+"-{FLASH ON}<
     {FLASH ON}":CHAR,30,14,M$       :rem  133
4100 GETZ$:IFZ$="{HOME}"THENM$=LEFT$(M$,1
     )+"{FLASH ON}<{FLASH ON}{ 3 SPAZI}":
     CHAR,30,14,M$:GOTO4070          :rem   77
4110 IFZ$<"A"ORZ$>"H"THEN4100        :rem  113
4120 M$=LEFT$(M$,3)+Z$+"{FLASH ON}<
     {FLASH ON}":CHAR,30,14,M$       :rem   84
4130 GETZ$:IFZ$="{HOME}"THENM$=LEFT$(M$,3
     )+"{FLASH ON}<{FLASH ON}{ 2 SPAZI}":
     CHAR,30,14,M$:GOTO4100          :rem   76
4140 IFZ$<"1"ORZ$>"8"THEN4130        :rem   87
4150 M$=LEFT$(M$,4)+Z$:CHAR,30,14,M$
                                     :rem  167
4160 F$(1)=LEFT$(M$,2):F$(2)=RIGHT$(M$,2)
                                     :rem  137
4170 FORI=1TO2:RESTORE9500           :rem  107
4180 READX$,Y1:IFY1=0THEN7000        :rem  209
4190 IFF$(I)=X$THENB(I)=Y1:GOTO4210
                                     :rem   38
4200 GOTO4180                        :rem  204
4210 NEXT                            :rem    6
4220 IFA(B(1))<>PBANDA(B(1))<>DBTHEN7000
                                     :rem  216
4230 IFA(B(2))<>BTHEN7000            :rem   97
4240 IFB(1)>B(2)ANDA(B(1))<>DBTHEN7000
                                     :rem   63
4250 IFR=0THEN4330                   :rem   18
```

```
4260 IFR<>1THEN4290                    :rem  86
4265 IFABS(B(1)-B(2))>7THEN4330 :rem 171
4270 CHAR,30,19,"{RVS}{CYN}PRESA":CHAR,28
     ,20,"OBBLIGATA"                :rem  29
4280 FORI=600TO700STEP4:SOUND1,I,1:NEXT
                                    :rem 186
4285 FORI=1TO500:NEXT:CHAR,30,19,"
     { 5 SPAZI}":CHAR,28,20,"{ 9 SPAZI}
     {OFF}":GOTO4020                :rem 198
4290 IFB(1)=XRANDABS(B(1)-B(2))>7THEN4330
                                    :rem  39
4300 CHAR,29,19,"{RVS}{CYN}O PRENDI":CHAR
     ,29,20,"O PASSI{OFF}"          :rem 146
4310 SOUND1,300,30:FORI=1TO900:NEXT
                                    :rem  93
4320 CHAR,29,19,"[< 9 K>]":CHAR,29,20,"
     [< 8 K>]":GOTO4020             :rem  91
4330 IFABS(B(1)-B(2))<=7THEN4370:rem 227
4340 IFA((B(1)+B(2))/2)<>PNANDA((B(1)+B(2
     ))/2)<>DNTHEN7000              :rem  55
4350 IFA((B(1)+B(2))/2)=DNANDA(B(1))<>DBT
     HEN7000                        :rem  65
4360 A((B(1)+B(2))/2)=B:MO=1:HS=HS+1
                                    :rem 145
4370 A(B(2))=A(B(1)):A(B(1))=B  :rem 199
4380 PRINT"{OFF}":GOSUB8100:IFHS=12THEN75
     00                             :rem 180
4390 IFMO=0THEN5000                 :rem  92
4400 CHAR,27,19,"{CYN}{RVS}ALTRA PRESA?":
     CHAR,30,20,"(S/N){OFF}":SOUND1,700,3
     0                              :rem  62
4410 GETZ$:IFZ$<>"N"ANDZ$<>"S"THEN4410
                                    :rem 212
4420 CHAR,27,19,"[< 12 K>]":CHAR,30,20,"
     [< 6 K>]"                      :rem 186
4430 IFZ$="S"THENR=2:XR=B(2):GOTO4020
                                    :rem 187
5000 :                              :rem 255
5010 REM MOSSA C16                  :rem 215
5020 :                              :rem   1
5030 CHAR,30,7,"{OFF}{CYN}[< 5 K>]":SOUND
     1,800,15:I=24                  :rem  14
5040 FORF=ITO72:IFA(F)<>PNANDA(F)<>DNTHEN
     5090                           :rem 245
5050 FORD=1TO4                      :rem  58
5060 IFA(F)<>DNANDD>2THEN5080       :rem 193
5070 IFA(F+2*X(D))=BAND(A(F+X(D))=PBOR(A(
     F+X(D))=DBANDA(F)=DN))THENQ=X(D):GOT
     O5100                          :rem  11
5080 NEXTD:I=F+1:GOTO5040           :rem 238
5090 NEXTF:GOTO5190                 :rem 149
5100 A(F+2*Q)=A(F):CS=CS+1:GOSUB5500
                                    :rem  14
5110 A(F+Q)=B:A(F)=B:P=F+2*Q        :rem  16
5120 GOSUB8100:IFCS=12THEN7700      :rem  13
5125 M=0:FORD=1TO4                  :rem  49
5130 IFA(P)<>DNANDD>2THEN5160       :rem 200
5140 IFA(P+2*X(D))=BAND(A(P+X(D))=PBOR(A(
     P+X(D))=DBANDA(P)=DN))THENM=X(D)
                                    :rem 244
5150 IFM=0THENNEXTD                 :rem 198
5160 IFM=0THENGOSUB5600:GOTO4000:rem 198
5170 A(P+M)=B:A(P+2*M)=A(P):A(P)=B
                                    :rem  80
5180 CS=CS+1:P=P+2*M:GOTO5120       :rem 185
5190 MS=0:PI=0:PM=0:I=23            :rem 133
5195 DOUNTILI=72                    :rem 226
5200 I=I+1:IFA(I)<>PNTHEN5280       :rem 172
5210 FORD=1TO2:LS=0                 :rem 124
5220 IFA(I+X(D))=BTHENLS=12:ELSEGOTO5270
                                    :rem 162
5230 IFI+X(D)<28THENPM=I:PI=I+X(D):MS=8:E
     XIT                            :rem 230
5235 IFA(I+X(D+4))=PBANDA(I-X(D+4))=BTHEN
     LS=LS+1                        :rem  43
5240 IFA(I+2*X(D))=PBORA(I+2*X(D))=DBTHEN
     LS=LS-2                        :rem  52
5245 IFI>68THENLS=LS-1              :rem  92
5247 IFA(I+X(D+4))=PNAND(A(I+2*(D+4))=PBO
     RA(I+2*X(D+4))=DB)THENLS=LS-3
                                    :rem 243
5250 IF(A(I-13)=PBORA(I-13)=DB)ANDA(I+2*X
     (D)+13)=BTHENLS=LS-2           :rem 240
5253 IFI<40ANDA(I+1)=PNANDA(I-1)=PNTHENLS
     =LS+3                          :rem  27
5255 IFA(I-13)=PBANDA(I+2*X(D)+13)<>BANDA
     (I+2*X(D))<>PBTHENLS=LS+.5 :rem 110
5258 IFA(I-X(D))=DBTHENLS=LS+1.5:rem 132
5259 IFI<68ANDA(I-X(D))=PNANDA(I-2*X(D))=
     DBTHENLS=LS-3                  :rem 235
5260 IFA(I-13)=BANDA(I+2*X(D)+13)=DBTHENL
     S=LS-2                         :rem 196
5265 GOSUB5900                      :rem  32
5270 NEXT:GOTO5350                  :rem  77
5280 IFA(I)<>DNTHEN5350             :rem  65
5290 FORD=1TO4:LS=0:IFD<3THENK=13:ELSEK=-
     13                             :rem 153
5300 IFA(I+X(D))=BTHENLS=11:ELSEGOTO5340
                                    :rem 158
5305 IFA(I+2*X(D))=PBTHENLS=LS+1:rem 128
5310 IFA(I+2*X(D))=DBTHENLS=LS-2:rem 115
5315 IFA(I+X(D+4))=DNANDA(I-X(D+4))=BTHEN
     LS=LS+1                        :rem  42
5320 IFA(I-K)=DBANDA(I+2*X(D)+K)=BTHENLS=
     LS-2                           :rem 143
5325 IFA(I-K)=BANDA(I+2*X(D)+K)=DBTHENLS=
     LS-2                           :rem 148
5330 IFA(I-K)=DBANDA(I+2*X(D)+K)=DBTHENLS
     =LS+3                          :rem 211
5335 GOSUB5900                      :rem  30
5340 NEXTD                          :rem  79
5350 LOOP                           :rem   7
5360 IFMS=0THEN7500                 :rem 101
5370 A(PI)=A(PM):A(PM)=B            :rem  78
5380 F=PM:GOSUB5500:P=PI:GOSUB5600
                                    :rem  89
5390 GOSUB8100:GOTO4000             :rem  81
5500 RESTORE9500:M$=""              :rem 232
5510 READX$,Y1:IFY1<>FTHEN5510      :rem  38
5520 M$="{RVS}{CYN}"+X$+"-":RETURN
                                    :rem 204
5600 RESTORE9500                    :rem 189
5610 READX$,Y1:IFY1<>PTHEN5610      :rem  50
5620 M$=M$+X$+"{OFF}":CHAR,30,7,M$
                                    :rem 123
5630 FORI=700TO800STEP4:SOUND1,I,1:NEXT
                                    :rem 188
5640 R=0:I=23:DOUNTILI=72           :rem 251
5650 I=I+1:IFA(I)<>PBANDA(I)<>DBTHEN5690
                                    :rem  92
5660 FORD=4TO1STEP-1:IFD<3ANDA(I)=PBTHEN5
     680                            :rem 214
5670 IFA(I+X(D))=PNANDA(I+2*X(D))=BTHENR=
     1:EXIT                         :rem  51
5675 IFA(I+X(D))=DNANDA(I)=DBANDA(I+2*X(D
     ))=BTHENR=1:EXIT               :rem 157
5680 NEXT                           :rem  18
5690 LOOP                           :rem  14
5700 RETURN                         :rem 172
```

```
5900 IFLS<MSTHENRETURN              :rem  231
5910 IFLS=MSANDRND(0)>.5THENRETURN
                                    :rem  194
5920 MS=LS:PM=I:PI=I+X(D):RETURN:rem   52
7000 :                              :rem    1
7010 REM MOSSA INVALIDA             :rem  119
7020 :                              :rem    3
7030 CHAR,30,19,"MOSSA":CHAR,28,20,"INVAL
     IDA!"                          :rem   85
7040 FORI=600TO400STEP-4:SOUND1,I,1:NEXT
                                    :rem  225
7050 FORI=1TO100:NEXT               :rem   23
7060 CHAR,30,19,"{ 5 SPAZI}":CHAR,28,20,"
     { 9 SPAZI}{OFF}"               :rem  254
7070 GOTO4020                       :rem  205
7500 :                              :rem    6
7510 REM FINE PARTITA               :rem  232
7520 :                              :rem    8
7530 CHAR,27,19,"{RVS}{CYN}HAI{ 2 SPAZI}
     VINTO!"                        :rem  188
7540 FORJ=1TO5:FORI=600TO800STEP4
                                    :rem   47
7550 SOUND1,I,1:NEXTI,J             :rem  149
7560 CHAR,26,21,"GIOCHI ANCORA":CHAR,30,2
     2,"(S/N){OFF}"                 :rem  132
7570 GETZ$:IFZ$="S"THENPRINT"{CLR}":GOTO2
     180                            :rem   29
7580 IFZ$<>"N"THEN7570              :rem  239
7590 PRINT"{CLR}":CHAR,16,10,"{WHT}C I A
     O"                             :rem   66
7600 POKE65298,4:POKE65299,208:END
                                    :rem  185
7700 CHAR,28,19,"{RVS}{CYN}HO VINTO!":GOT
     O7540                          :rem  196
8000 :                              :rem    2
8010 REM DA PEDINA A DAMA           :rem   55
8020 :                              :rem    4
8030 FORI=24TO72                    :rem  170
8040 IFA(I)=PBANDI>68THENA(I)=DB:rem  154
8050 IFA(I)=PNANDI<28THENA(I)=DN:rem  173
8060 NEXT:RETURN                    :rem   39
8100 :                              :rem    3
8110 REM AGGIORNAMENTO SCACCHIERA
                                    :rem   63
8120 :                              :rem    5
8130 GOSUB8000:RESTORE9500          :rem   64
8140 FORI=0TO19STEP6:FORJ=4TO23STEP6
                                    :rem   63
8150 READX$,Y1:GOSUB8200:CHAR,J,I,Z0$:NEX
     T                              :rem   54
8160 FORJ=1TO20STEP6:READX$,Y1:GOSUB8200
                                    :rem  241
8170 CHAR,J,I+3,Z0$:NEXT:NEXT:RETURN
                                    :rem   29
8200 IFA(Y1)=PNTHENZ0$=Z2$:GOTO8250
                                    :rem   92
8210 IFA(Y1)=PBTHENZ0$=Z3$:GOTO8250
                                    :rem   82
8220 IFA(Y1)=DNTHENZ0$=Z5$:GOTO8250
                                    :rem   85
8230 IFA(Y1)=DBTHENZ0$=Z6$:GOTO8250
                                    :rem   75
8240 IFA(Y1)=BTHENZ0$=Z4$           :rem  196
8250 RETURN                         :rem  175
9500 :                              :rem    8
9510 REM DATA CASELLE               :rem  198
9520 :                              :rem   10
9530 DATA"B8",72,"D8",71,"F8",70,"H8",69
                                    :rem  202
9540 DATA"A7",66,"C7",65,"E7",64,"G7",63
                                    :rem  198
9550 DATA"B6",59,"D6",58,"F6",57,"H6",56
                                    :rem  207
9560 DATA"A5",53,"C5",52,"E5",51,"G5",50
                                    :rem  176
9570 DATA"B4",46,"D4",45,"F4",44,"H4",43
                                    :rem  185
9580 DATA"A3",40,"C3",39,"E3",38,"G3",37
                                    :rem  181
9590 DATA"B2",33,"D2",32,"F2",31,"H2",30
                                    :rem  163
9600 DATA"A1",27,"C1",26,"E1",25,"G1",24
                                    :rem  159
9610 DATA"0",0                      :rem  186
```

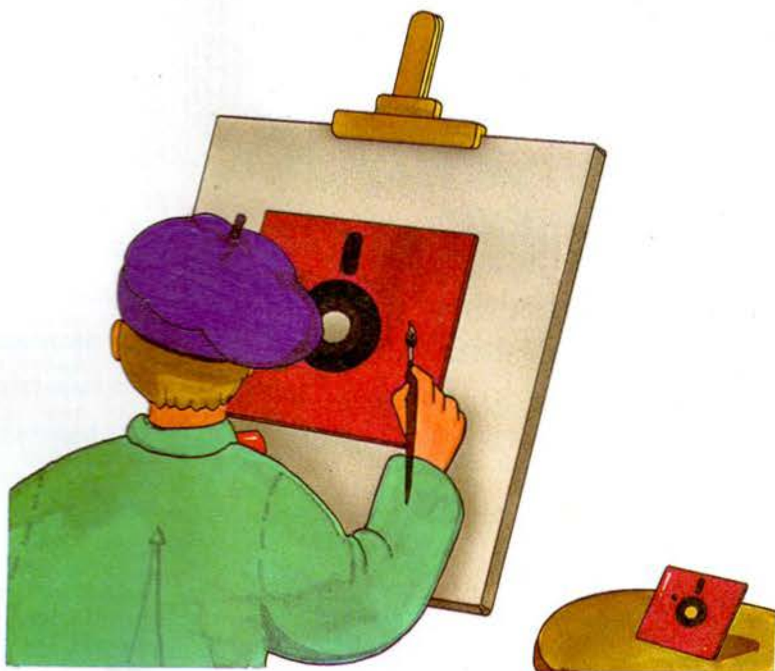


Turbocopy per C64

di **K. Winters**
trad. ed adatt.
di **L. Priotto**

Chi lavora abitualmente con il proprio C64 ed il disk drive sente spesso la necessità di crearsi alcune copie di sicurezza dei propri dischi, vuoi per non rischiare di perdere dei file importanti, vuoi per poter riformattare il disco "di lavoro" ed usarlo quindi nuovamente.

TURBOCOPY, un programma completamente in linguaggio macchina, è una eccezionale utility, che permette di duplicare interamente un dischetto in soli tre passaggi.
Il suo utilizzo è molto semplice: dopo averlo caricato ed attivato vi verrà chiesto di premere un tasto per iniziare. Fatto questo il programma provvede a posizionare correttamente la testina del drive (anche se il rumore è molto simile, non abbiate paura: non sta formattando!) e quindi vi chiede di inserire nel drive il disco sorgente (cioè il disco che desiderate copiare). Fatelo e premete un tasto qualsiasi: lo schermo scomparirà e la spia del drive inizierà a lampeggiare. Non allarmatevi, in quanto non si è verificata alcuna condizione di errore! TURBOCOPY copia traccia per traccia e settore per settore, con una serie di veloci accessi al disco, che provocano questo lampeggiamento.
Dopo pochi secondi ricomparirà lo schermo, ed il programma vi chiederà di inserire il disco destinazione (quello sul quale desiderate fare la copia).
Quest'ultimo può anche essere completamente vergine e non formattato, in quanto TURBOCOPY provvede automaticamente alla formattazione mano a mano che scrive sul dischetto.
Dopo aver inserito il disco destinazione premete un tasto qualsiasi e nuovamente lo schermo scomparirà, mentre la luce rossa del drive inizierà a lampeggiare: sempre tutto normale!
Questa procedura va ripetuta solamente tre volte, fino a quando il programma avvertirà che la copia è stata completata. A questo punto, premendo un tasto, reinizializzerete TURBOCOPY, preparandolo per un'altra copia. Poiché TURBOCOPY non prevede un'uscita software dal programma, per disattivarlo occorre spegnere il computer.



Eventuali errori in lettura o in scrittura vengono segnalati indicando la traccia e il settore sul quale è presente l'errore, ed il suo tipo (per una legenda di questi codici di errore si veda il manuale del drive o la serie di articoli "Iniziare con un sistema a dischi", da noi pubblicata lo scorso anno).
TURBOCOPY è uno dei più veloci programmi di back-up per 1541 attualmente esistenti (un back-up completo richiede soltanto circa tre minuti!) che va ad aggiungersi alle ormai numerose utility della collezione di SUPERCOMMODORE.
Il principio di funzionamento di TURBOCOPY si basa sulla bufferizzazione dei dati: viene creato, all'interno del C64, un buffer di circa 55 Kbyte, che viene riempito ad ogni lettura dal primo disco e riscritto quindi sul secondo.
Poiché un disco contiene fino a 170 Kbyte di dati, occorrono tre diverse "passate" per poterlo duplicare interamente.
È superfluo sottolineare che qualsiasi dato eventualmente presente sul disco destinazione andrà distrutto: TURBOCOPY produce infatti una copia esatta del disco sorgente, compreso nome del disco e codice identificatore.

Note per il caricamento e la digitazione

Se si dispone della cassetta allegata alla rivista, è sufficiente posizionarsi con il nastro in corrispondenza dell'inizio del programma e digitare:

LOAD

Il programma andrà quindi mandato in esecuzione digitando:

RUN

Ricordiamo che per conoscere il punto esatto di inizio dei vari programmi su cassetta è sufficiente inserire quest'ultima completamente riavvolta nel registratore, azzerare il contanastro e digitare:

LOAD"*"

Dal momento che sicuramente non esiste sulla cassetta alcun programma di nome "*", il computer passerà in rassegna tutti i programmi incontrati, visualizzandone il nome con il messaggio:

FOUND XXXX

Sarà quindi utile annotarsi su un foglio i nomi dei programmi e la relativa posizione del contanastro (sottrarre tre o quattro unità dal valore corrispondente all'apparizione del messaggio FOUND...).
Gli abbonati alla sola rivista dovranno invece usare il programma MLX (vedere nelle pagine verdi) per digitare il listato in linguaggio macchina. La procedura da seguire, in questo caso, è:

1. Spegnere e riaccendere il computer e digitare:

POKE44,32:POKE32*256,0:NEW [RETURN]

2. Caricare in memoria MLX (che sarà stato precedentemente digitato e salvato) e rispondere alle sue richieste circa gli indirizzi iniziale e finale con:

indirizzo iniziale: 2049
indirizzo finale : 4592

3. Dopo avere digitato e salvato il programma tramite MLX si potranno seguire per il caricamento le istruzioni date più sopra per i possessori di cassetta.

Turbocopy per C64

```
2049 :013,008,010,000,158,032,222
2055 :050,048,054,051,032,000,242
2061 :000,000,076,026,008,000,123
2067 :003,000,013,226,017,108,130
2073 :00[illegible],173,018,008,133,253,104
2079 :173,019,008,133,254,173,023
2085 :020,008,133,251,173,021,131
2091 :008,133,252,056,173,022,175
2097 :008,237,020,008,133,249,192
2103 :173,023,008,237,021,008,013
2109 :133,250,032,231,255,169,107
2115 :015,162,008,168,032,186,126
2121 :255,169,000,032,189,255,205
2127 :032,192,255,056,165,249,004
2133 :233,034,133,249,165,250,125
2139 :233,000,133,250,144,004,087
2145 :169,034,208,012,165,249,166
2151 :073,255,133,248,169,034,247
2157 :229,248,240,085,133,248,012
2163 :162,015,032,201,255,169,181
2169 :077,032,210,255,169,045,141
2175 :032,210,255,169,087,032,144
2181 :210,255,165,253,032,210,234
2187 :255,165,254,032,210,255,030
2193 :165,248,032,210,255,160,191
2199 :000,177,251,032,210,255,052
2205 :200,196,248,208,246,032,007
2211 :204,255,192,034,144,029,253
2217 :024,165,251,105,034,133,113
2223 :251,165,252,105,000,133,057
2229 :252,024,165,253,105,034,246
2235 :133,253,165,254,105,000,073
2241 :133,254,076,082,008,162,140
2247 :015,032,201,255,169,077,180
2253 :032,210,255,169,045,032,180
2259 :210,255,169,069,032,210,132
2265 :255,173,024,008,032,210,151
2271 :255,173,025,008,032,210,158
2277 :255,032,204,255,032,018,001
2283 :011,147,017,032,032,032,250
2289 :084,085,082,066,079,067,192
2295 :079,080,089,032,068,073,156
2301 :032,083,085,080,069,082,172
2307 :067,079,077,077,079,068,194
2313 :079,082,069,013,013,032,041
2319 :032,032,032,032,066,065,018
2325 :067,075,085,080,032,049,153
2331 :053,052,049,032,073,078,108
2337 :032,051,032,080,065,083,120
2343 :083,073,013,032,032,032,048
2349 :032,032,032,032,070,079,066
2355 :082,077,065,084,084,065,252
2361 :032,065,085,084,079,077,223
2367 :065,084,073,067,065,077,238
2373 :069,078,084,069,013,013,139
2379 :032,018,080,082,069,077,177
2385 :073,032,085,078,032,084,209
2391 :065,083,084,079,046,046,234
2397 :046,032,032,032,013,013,005
2403 :000,032,018,011,013,032,205
2409 :073,078,083,069,082,073,051
2415 :083,067,073,032,068,073,251
2421 :083,067,079,032,083,079,028
2427 :082,071,046,032,000,032,130
2433 :200,010,169,165,032,056,249
2439 :011,169,000,162,016,133,114
2445 :003,134,004,032,106,011,175
2451 :240,031,008,160,000,145,219
2457 :003,230,003,040,048,012,233
2463 :032,106,011,145,003,200,144
2469 :208,248,230,004,208,229,012
2475 :041,031,170,202,134,006,243
2481 :076,097,010,032,106,011,253
2487 :016,007,041,063,133,005,192
2493 :076,144,009,032,161,010,109
2499 :032,018,011,013,032,018,063
2505 :073,078,083,069,082,073,147
2511 :083,067,073,032,068,073,091
2517 :083,067,079,032,068,069,099
2523 :083,084,046,000,032,200,152
2529 :010,032,056,011,169,000,247
2535 :133,003,169,016,133,004,177
```

```
2541 :160,000,032,106,011,208,242
2547 :023,177,003,008,032,056,030
2553 :011,230,003,040,048,240,053
2559 :177,003,032,056,011,200,222
2565 :208,248,230,004,208,228,107
2571 :016,015,041,063,133,005,028
2577 :165,003,166,004,133,048,024
2583 :134,049,076,239,009,201,219
2589 :003,208,006,032,161,010,193
2595 :076,100,009,201,009,208,126
2601 :011,165,048,166,049,133,101
2607 :003,134,004,076,239,009,000
2613 :201,036,208,003,076,156,221
2619 :011,032,161,010,032,018,067
2625 :011,013,032,032,087,082,066
2631 :073,084,069,032,069,082,224
2637 :082,079,082,033,084,082,007
2643 :065,067,067,073,065,000,164
2649 :165,005,032,171,010,076,036
2655 :156,011,169,054,133,001,107
2661 :032,018,011,013,032,032,239
2667 :082,069,065,068,032,069,236
2673 :082,082,079,082,033,084,043
2679 :082,065,067,067,073,065,026
2685 :000,165,005,032,171,010,252
2691 :032,018,011,032,083,069,120
2697 :084,084,079,082,069,000,023
2703 :165,006,032,171,010,169,184
2709 :013,032,210,255,169,052,112
2715 :133,001,120,076,144,009,126
2721 :169,054,133,001,169,027,202
2727 :141,017,208,096,248,170,023
2733 :240,008,169,000,024,105,207
2739 :001,202,208,250,216,072,104
2745 :074,074,074,074,032,193,194
2751 :010,104,041,015,009,048,162
2757 :076,210,255,088,169,037,008
2763 :141,015,212,169,000,141,113
2769 :020,212,169,043,141,019,045
2775 :212,169,047,141,024,212,252
2781 :169,052,141,023,212,169,219
2787 :032,141,018,212,169,033,064
2793 :141,018,212,032,228,255,095
2799 :208,013,173,028,212,073,178
2805 :089,141,022,212,208,241,134
2811 :076,226,010,169,000,141,105
2817 :018,212,169,011,141,017,057
2823 :208,160,032,202,208,253,046
2829 :136,208,250,120,096,104,159
2835 :133,053,104,133,054,160,144
2841 :000,230,053,208,002,230,236
2847 :054,177,053,240,011,032,086
2853 :210,255,230,053,208,245,214
2859 :230,054,208,241,230,053,035
2865 :208,002,230,054,108,053,192
2871 :000,133,107,169,053,133,138
2877 :001,169,011,141,000,221,092
2883 :173,000,221,016,251,169,129
2889 :003,141,000,221,162,004,092
2895 :169,003,070,107,106,070,092
2901 :107,106,074,074,234,141,053
2907 :000,221,202,208,239,162,099
2913 :001,202,208,253,169,052,214
2919 :133,001,096,169,053,133,176
2925 :001,169,011,141,000,221,140
2931 :173,000,221,016,251,169,177
2937 :003,141,000,221,162,005,141
2943 :202,234,208,252,162,004,165
2949 :173,000,221,010,008,010,043
2955 :038,107,040,038,107,202,159
2961 :208,242,169,052,133,001,182
2967 :165,107,073,255,096,032,111
2973 :161,010,032,018,011,013,146
2979 :013,032,032,067,079,080,210
2985 :073,065,032,067,079,077,050
2991 :080,076,069,084,065,033,070
2997 :013,013,000,032,200,010,193
3003 :032,161,010,076,233,008,195
3009 :255,255,255,255,255,255,187
3015 :255,255,255,255,255,255,193
3021 :255,255,255,255,255,255,199
3027 :255,255,255,255,255,255,205
3033 :255,255,255,255,255,255,211
3039 :255,255,255,255,255,255,217
3045 :255,255,255,255,255,255,223
3051 :255,255,255,255,255,255,229
3057 :255,255,255,255,255,255,235
3063 :255,255,255,255,255,255,241
3069 :255,255,255,255,255,255,247
3075 :255,255,255,255,255,255,253
3081 :255,255,255,255,255,255,003
3087 :255,255,255,255,255,255,009
3093 :255,255,255,255,255,255,015
3099 :255,255,255,255,255,255,021
3105 :255,255,255,255,255,255,027
3111 :255,255,255,255,255,255,033
3117 :255,255,255,255,255,255,039
3123 :255,255,255,255,255,255,045
3129 :255,255,255,255,255,255,051
3135 :255,255,255,255,255,255,057
3141 :255,255,255,255,255,255,063
3147 :255,255,255,255,255,255,069
3153 :255,255,255,255,255,255,075
3159 :255,255,255,255,255,255,081
3165 :255,255,255,255,255,255,087
3171 :255,255,255,255,255,255,093
3177 :255,255,255,255,255,255,099
3183 :255,255,255,255,255,255,105
3189 :255,255,255,255,255,255,111
3195 :255,255,255,255,255,255,117
3201 :255,255,255,255,255,255,123
3207 :255,255,255,255,255,255,129
3213 :255,255,255,255,255,255,135
3219 :255,255,255,255,255,255,141
```

```
3225 :255,255,255,255,255,255,147
3231 :255,255,255,255,255,255,153
3237 :255,255,255,255,255,255,159
3243 :255,255,255,255,255,255,165
3249 :255,255,255,255,255,255,171
3255 :255,255,255,255,255,255,177
3261 :255,255,255,255,255,255,183
3267 :255,255,255,255,255,255,189
3273 :255,255,255,255,255,255,195
3279 :255,255,255,255,255,255,201
3285 :255,255,255,255,255,255,207
3291 :255,255,255,255,255,255,213
3297 :255,255,255,255,255,255,219
3303 :255,255,255,255,255,255,225
3309 :255,255,255,255,255,255,231
3315 :255,255,255,255,255,255,237
3321 :255,255,255,255,255,255,243
3327 :255,165,000,041,006,240,194
3333 :017,201,002,208,003,076,000
3339 :026,003,201,004,208,003,200
3345 :076,197,005,076,078,004,197
3351 :076,158,253,165,068,074,049
3357 :074,074,074,074,170,189,172
3363 :074,004,133,012,165,119,030
3369 :208,004,230,119,165,067,066
3375 :133,010,169,005,133,009,250
3381 :169,090,133,075,169,082,003
3387 :133,036,032,086,245,080,159
3393 :254,184,173,001,028,197,134
3399 :036,240,009,198,075,208,069
3405 :239,169,010,076,105,249,157
3411 :162,000,080,254,184,173,168
3417 :001,028,149,037,232,224,248
3423 :007,144,243,032,151,244,148
3429 :165,022,069,023,069,024,217
3435 :069,025,069,026,240,007,031
3441 :198,009,208,192,076,030,058
3447 :244,165,024,197,006,240,227
3453 :003,076,011,244,133,034,114
3459 :165,022,166,023,133,018,146
3465 :134,019,166,025,232,232,177
3471 :228,067,144,002,162,000,234
3477 :169,002,133,049,076,061,127
3483 :004,202,208,021,076,025,179
3489 :004,134,007,134,025,138,091
3495 :069,024,069,023,069,022,187
3501 :133,026,032,052,249,162,059
3507 :090,032,086,245,080,254,198
3513 :184,173,001,028,217,036,056
3519 :000,208,218,200,192,008,249
3525 :208,240,032,086,245,080,064
3531 :254,184,173,001,028,145,220
3537 :048,200,208,245,160,186,232
3543 :080,254,184,173,001,028,167
3549 :153,000,001,200,208,244,003
3555 :032,224,248,165,056,197,125
3561 :071,208,045,032,233,245,043
3567 :197,058,240,011,166,007,150
3573 :189,080,001,208,035,169,159
3579 :005,208,033,230,025,165,149
3585 :025,032,094,005,185,000,086
3591 :002,032,094,005,200,208,036
3597 :247,173,000,028,073,008,030
3603 :141,000,028,169,255,044,144
3609 :169,132,166,007,009,128,124
3615 :157,080,001,008,160,000,181
3621 :140,254,002,040,016,007,240
3627 :198,010,208,003,076,158,184
3633 :253,138,024,101,012,197,006
3639 :067,144,002,229,067,170,222
3645 :189,080,001,016,005,232,072
3651 :138,076,054,004,076,162,065
3657 :003,009,009,010,011,173,032
3663 :012,028,041,031,009,192,136
3669 :141,012,028,169,255,141,063
3675 :003,028,169,085,141,001,006
3681 :028,165,068,074,074,074,068
3687 :074,074,170,189,190,005,037
3693 :133,012,160,000,132,010,044
3699 :162,000,165,057,153,000,140
3705 :002,200,200,165,010,153,083
3711 :000,002,200,165,006,153,141
3717 :000,002,200,165,019,153,160
3723 :000,002,200,165,018,153,165
3729 :000,002,200,169,015,153,172
3735 :000,002,200,153,000,002,252
3741 :200,185,250,001,089,251,109
3747 :001,089,252,001,089,253,080
3753 :001,153,249,001,230,010,045
3759 :165,010,197,067,144,192,182
3765 :152,072,169,002,133,049,246
3771 :032,048,254,104,168,136,161
3777 :185,000,002,153,069,002,092
3783 :136,208,247,173,000,002,197
3789 :141,069,002,032,245,253,179
3795 :169,000,133,050,169,255,219
3801 :141,001,028,162,005,080,122
3807 :254,184,202,208,250,162,203
3813 :010,164,050,080,254,184,203
3819 :185,000,002,141,001,028,080
3825 :200,202,208,243,162,009,241
3831 :080,254,184,169,085,141,136
3837 :001,028,202,208,245,169,082
3843 :255,162,005,080,254,184,175
3849 :141,001,028,202,208,247,068
3855 :169,085,080,254,184,141,160
3861 :001,028,162,188,080,254,222
3867 :184,169,221,141,001,028,003
3873 :232,208,245,160,000,080,190
3879 :254,184,169,221,141,001,241
3885 :028,136,208,245,169,085,148
3891 :166,012,080,254,184,141,120
3897 :001,028,202,208,247,165,140
3903 :050,024,105,010,133,050,179
```

```
3909 :198,010,208,142,080,254,193
3915 :184,080,254,184,032,000,041
3921 :254,169,000,141,254,002,133
3927 :133,080,169,001,076,105,139
3933 :249,133,008,044,000,024,039
3939 :016,251,169,016,141,000,180
3945 :024,044,000,024,048,251,240
3951 :162,004,169,000,006,008,204
3957 :042,010,006,008,042,010,235
3963 :141,000,024,202,208,240,170
3969 :162,001,202,208,000,234,168
3975 :234,169,015,141,000,024,206
3981 :096,044,000,024,016,251,060
3987 :169,016,141,000,024,044,029
3993 :000,024,048,251,162,004,130
3999 :202,208,253,142,000,024,220
4005 :162,004,173,000,024,074,090
4011 :008,074,074,102,008,040,221
4017 :102,008,202,208,241,169,083
4023 :015,141,000,024,165,008,024
4029 :096,008,011,016,007,076,147
4035 :093,006,165,067,133,010,157
4041 :032,000,254,169,000,032,176
4047 :094,005,234,032,142,005,207
4053 :048,235,170,202,134,007,241
4059 :169,002,133,049,160,000,220
4065 :032,142,005,153,000,002,047
4071 :200,208,247,032,233,245,116
4077 :133,058,032,143,247,173,255
4083 :254,002,133,124,169,000,157
4089 :141,254,002,133,080,165,000
4095 :015,240,003,076,172,007,000
4101 :032,016,245,162,009,080,037
4107 :254,184,202,208,250,169,254
4113 :255,141,003,028,173,012,117
4119 :028,041,031,009,192,141,209
4125 :012,028,169,255,162,005,148
4131 :141,001,028,184,080,254,211
4137 :184,202,208,250,160,187,208
4143 :185,000,001,080,254,184,239
4149 :141,001,028,200,208,244,107
4155 :165,124,141,254,002,177,154
4161 :048,080,254,184,141,001,005
4167 :028,200,208,245,080,254,062
4173 :032,000,254,173,000,028,052
4179 :073,008,141,000,028,169,246
4185 :000,141,254,002,198,010,182
4191 :240,003,076,204,005,076,187
4197 :158,253,120,169,007,133,173
4203 :071,162,000,202,208,253,235
4209 :169,001,133,006,169,176,255
4215 :032,163,007,169,224,032,234
4221 :163,007,032,142,005,165,127
4227 :006,133,013,169,000,032,228
4233 :094,005,165,006,009,128,032
4239 :032,094,005,032,152,007,209
4245 :169,000,133,120,133,131,067
4251 :133,119,133,016,169,226,183
4257 :032,163,007,201,002,144,198
4263 :022,165,016,208,018,169,253
4269 :192,032,163,007,169,176,144
4275 :032,163,007,230,016,169,028
4281 :000,133,120,133,131,162,096
4287 :000,134,009,189,080,001,092
4293 :201,255,208,008,232,228,049
4299 :067,208,244,076,240,006,020
4305 :164,120,185,219,254,240,111
4311 :010,169,000,157,080,001,120
4317 :230,009,076,201,006,134,109
4323 :007,232,138,009,128,032,005
4329 :094,005,166,007,076,201,014
4335 :006,165,009,240,023,133,047
4341 :119,165,131,056,249,219,160
4347 :254,133,131,185,219,254,147
4353 :088,032,118,214,120,230,035
4359 :120,076,159,006,165,131,152
4365 :088,032,118,214,120,230,047
4371 :006,165,006,201,012,240,137
4377 :011,201,023,240,007,201,196
4383 :036,240,003,076,134,006,014
4389 :169,000,032,094,005,169,250
4395 :001,032,094,005,234,032,185
4401 :142,005,165,013,133,006,001
4407 :169,176,032,163,007,165,255
4413 :006,009,128,032,094,005,079
4419 :169,005,133,014,169,009,054
4425 :032,094,005,169,230,032,123
4431 :163,007,169,228,162,000,040
4437 :134,015,032,163,007,201,125
4443 :002,176,016,169,009,032,239
4449 :094,005,230,015,169,228,070
4455 :032,163,007,201,002,144,140
4461 :009,198,014,208,213,169,152
4467 :005,076,146,007,230,006,073
4473 :165,006,201,012,240,011,244
4479 :201,023,240,007,201,036,067
4485 :240,011,076,060,007,169,184
4491 :003,032,094,005,076,127,220
4497 :006,032,094,005,076,108,210
4503 :006,169,000,162,020,157,153
4509 :080,001,202,016,250,096,034
4515 :133,000,088,165,000,048,085
4521 :252,120,096,032,010,245,156
4527 :160,187,185,000,001,080,020
4533 :254,184,077,001,028,208,165
4539 :026,200,208,242,165,124,128
4545 :141,254,002,177,048,080,127
4551 :254,184,077,001,028,208,183
4557 :008,200,192,253,208,241,027
4563 :076,080,006,169,000,141,171
4569 :254,002,076,197,246,255,223
4575 :255,000,000,000,000,000,222
4581 :000,000,000,000,000,000,229
4587 :016,080,000,255,000,000,074
```

PRINT

Recensioni Libri

di **M. Cristuib Grizzi**

I libri italiani vengono offerti con il 10% di sconto sul prezzo di copertina.

TECNICHE DI PROGRAMMAZIONE SUL COMMODORE 64

di **D. Lawrence**
Gruppo Editoriale Jackson



Un libro diverso dalle solite raccolte di programmi e da quei volumi che pretendono di migliorare la capacità di programmazione del lettore presentando banali routine di conversione tra gradi Fahrenheit e Centigradi, o tra piedi e metri.
Il libro è infatti dedicato a tutte quelle persone che hanno già iniziato a percorrere i primi passi nella programmazione, ma che trovano difficoltà nel creare programmi poco più che dilettanteschi.
Vengono presentati suggerimenti, trucchi, algoritmi ed informazioni tali da permettere un vero "salto di qualità" nella produzione di propri programmi, siano essi dei giochi o degli applicativi.
In particolare per la creazione di questi ultimi vengono fornite spiegazioni e routine "pronte per l'uso", per raggiungere obbiettivi semplici (come allineare i decimali di una serie di numeri tabulati), oppure complessi (come l'inserimento di dati ad alta velocità in grosse matrici).
I metodi forniti per l'impostazione e la stesura dei programmi si rifanno alle teorie della modularità, e vengono rivelati sistemi per rilevare, gestire e rintracciare gli errori, insieme a come ottenere il massimo dalle caratteristiche di velocità, memoria, funzioni e gestione dati del C64.
Tutte le routine presentate sono listate in forma molto leggibile, e non contengono caratteri grafici o di controllo che possano inficiare la comprensione delle linee BASIC.

COMMODORE 16: ESSENZIALI E ROUTINE

di **D. Lawrence**
Gruppo Editoriale Jackson

Sta iniziando a diffondersi anche in Italia la letteratura sul nuovo C16, e già si trovano alcuni libri che trattano gli argomenti più interessanti, dai giochi alle utility. Questo libro, in particolare, si propone di offrire allo stesso tempo una raccolta di programmi – contenuti nella cassetta allegata – che è possibile adattare per i propri scopi, un elenco di subroutine che è possibile aggiungere ai propri programmi ed un'introduzione al linguaggio BASIC del C16.
Diviso in cinque capitoli, tratta rispettivamente nel primo ("Esperimenti con l'ora") come utilizzare il clock interno del C16, nel secondo ("Dipingere con i numeri") come creare grafici ad istogrammi o diagrammi a torta, nel terzo ("Suoni e luci") come generare disegni in alta e bassa risoluzione e come elaborare e riascoltare motivi musicali. Il quarto capitolo ("Sempre più seriamente") presenta un piccolo archivio, un programma con funzioni di "dizionario" per nomi e numeri ed un word processor realizzato in BASIC.
L'ultimo capitolo ("Questioni di soldi") contiene un programma per l'aggiornamento del proprio conto corrente bancario ed un altro programma per gestire una piccola serie di conti nella forma consueta di debiti e crediti. I programmi, contenuti interamente sulla cassetta, sono invece spezzettati all'interno del libro in forma di vari "moduli", ognuno dei quali separatamente commentato ed analizzato, ed i listati sono realizzati con una particolare codifica, che elimina tutti i caratteri grafici o di controllo, migliorandone la leggibilità complessiva.



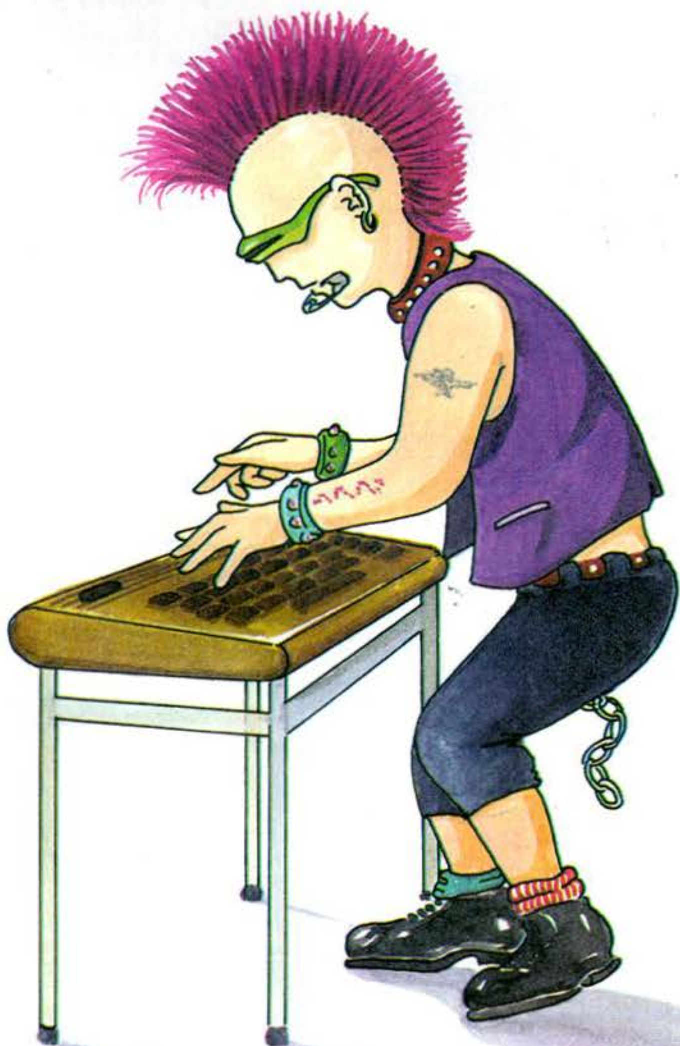
I libri recensiti possono essere ordinati utilizzando il coupon di pag. 82

VIC-sound

Un piano...elettronico per C64

di **P. Ponzo**
trad. ed adatt.
di **C. Stefani**

Ecco un programma allo stesso tempo utile per chi conosce la musica e divertente per chi invece si limita a "strimpellare" qualcosa.



Una volta caricato – attraverso l'apposito programmino caricatore – e mandato in esecuzione il programma principale, verrà visualizzata la tastiera di un pianoforte ed il computer domanderà se si desidera che venga suonato un brano introduttivo. In caso affermativo due veloci mani da pianista si avvicineranno alla tastiera e si produrranno nell'esecuzione di un noto brano, già memorizzato all'interno del programma stesso.
Vi saranno quindi offerte le opzioni di caricare un brano da disco, caricarlo da nastro, iniziare a suonare ed uscire dal programma. Dal momento che non disporrete di un brano già registrato su nastro o disco, la cosa più logica sarà iniziare a suonare, premendo i tasti contrassegnati dai caratteri visualizzati sullo schermo. A mano a mano che procederete nella pressione dei tasti, verranno visualizzate sul video le mani che premono gli stessi tasti sulla tastiera del piano. Potete correggere eventuali errori premendo il tasto "–", che provoca il ritorno indietro di una nota. Segnalate infine al computer che avete terminato di suonare, premendo il tasto ".".
Verrà così visualizzato un secondo menu di selezione, con le seguenti opzioni:

1 suonare
2 aggiungere note

3 correggere note
4 save su disco
5 save su nastro
6 cambiare il brano di introduzione
7 cambiare voce
8 opzione libera per personalizzazioni

Selezionando l'opzione 1, si riascolterà il brano precedentemente composto o caricato da nastro o disco; l'opzione 2 permette di aggiungere note ad un brano precedentemente composto o caricato da nastro o disco, mentre l'opzione 3 permette di correggere singole note stonate o comunque errate all'interno del brano.
Le opzioni 4 e 5 non hanno bisogno di commenti, mentre la 6 permette di sostituire il brano suonato dal computer all'inizio del programma con un altro brano, precedentemente composto o caricato da nastro o disco, e di inserirlo automaticamente all'interno del programma stesso. L'opzione 9 permette di usare alternativamente una delle tre diverse voci del C64, contraddistinta ognuna da una diversa forma d'onda; l'opzione 9, infine, è già pronta e lasciata libera per chi intendesse aggiungere al programma proprie e particolari funzioni.
Caricando un brano da disco, verrà dapprima mostrato un elenco dei brani attualmente presenti sul disco stesso (riconoscibili dal programma attraverso uno spazio vuoto come primo carattere del nome del file), e quindi verrà richesto il nome stesso. Si noti a questo proposito che per caricare un brano da nastro è obbligatorio digitare il nome del file, in quanto quest'ultimo viene registrato in forma sequenziale, come si nota dal lampeggiamento discontinuo dello schermo durante il caricamento da nastro.
Dal momento che i dati per gli SPRITE costituenti la forma delle mani vengono memorizzati nella parte bassa dell'area BASIC, il programma non funzionerà, se non caricato attraverso l'apposito programmino caricatore. Le prime linee del programma principale hanno appunto la funzione di "accorgersi" se il caricamento è avvenuto correttamente, e in caso contrario di visualizzare un messaggio e fermare l'esecuzione del programmma stesso.

Programma caricatore

```
4 POKE53280,4:POKE53281,0          :rem 141
5 POKE198,5:POKE631,19:POKE632,13:POKE633
  ,141:POKE634,131:POKE635,13      :rem 185
10 PRINT"{CLR}POKE44,12:POKE3072,0:CLR"
                                   :rem 151
20 PRINT"{ 8 GIU'}ATTENDERE MENTRE CARICO
    IL PROGRAMMA..."               :rem 152
```

Piano per C64

```
8 IFPEEK(44)=12THEN10              :rem 213
9 PRINT"QUESTO PROGRAMMA DEVE ESSERE CARI
  CATO{ 3 SPAZI}DAL SUO CARICATORE!!!":EN
  D                                :rem 219
10 PRINT"{CLR}":POKE53281,15:POKE53280,12
  :PRINT"{BLK}"                    :rem 88
20 DIMN(255),P(255)                :rem 224
30 GOSUB10000:REM* INIZIALIZZA     :rem 106
40 GOSUB1900: REM* TITOLO INIZIALE
                                   :rem 38
50 GOSUB1000: REM* TASTIERA PIANO
                                   :rem 194
60 IFNO=0THENGOSUB50000:REM* MUSICA
                                   :rem 173
70 GOTO2800:  REM* MENU #1         :rem 59
80 GOSUB1100: REM* LETTERE TASTIERA
                                   :rem 100
90 GOSUB2500: REM* ISTRUZIONI      :rem 24
115 GOSUB2200:REM* INSERZIONE NOTE
                                   :rem 95
120 GOSUB1400:REM* INPUT NOTE      :rem 230
130 GOSUB2000:REM* CANCELLA LETTERE
                                   :rem 102
140 GOSUB1200:REM* SUONA BRANO     :rem 24
150 GOSUB500: REM* MENU #2         :rem 125
160 GOTO2800                       :rem 154
200 REM INIZ.TUTTE LE VOCI         :rem 54
201 REM"[< 16 U>]                  :rem 25
205 POKESID+24,15:IFN>LKTHENRETURN
                                   :rem 196
210 POKEPA(VC),FR(N)/256:POKEPB(VC),FR(N)
    AND255                         :rem 35
220 POKESID+4+7*VC,WV(VC):PV=VC:VC=VC+1:I
    FVC=3THENVC=0                  :rem 172
225 RETURN                         :rem 121
350 REM INIZ. SID                  :rem 196
351 REM"[< 9 U>]                   :rem 23
352 SID=54272:RESTORE              :rem 25
353 FORX=0TO2                      :rem 28
354 READ AT(X),DE(X),SU(X),RE(X):rem 61
355 READ PW(X),WV(X)               :rem 139
356 AD(X)=AT(X)*16+DE(X)           :rem 53
357 SR(X)=SU(X)*16+RE(X)           :rem 119
358 NEXT X:RE=0                    :rem 117
359 DATA 00,00,15,11,1000,17       :rem 72
360 DATA 00,00,15,11,1000,17       :rem 64
361 DATA 00,00,15,11,1000,17       :rem 65
362 FORX=0TO2:Y=7*X:IF RE<RE(X)THENRE=RE(
    X)                             :rem 196
363 POKESID+5+Y,AD(X)              :rem 233
364 POKESID+6+Y,SR(X)              :rem 11
365 POKESID+2+Y,PW(X)/256          :rem 214
366 POKESID+3+Y,PW(X)AND255        :rem 123
367 POKESID+4+Y,8:REM RESET OSCILLATORE
                                   :rem 216
368 POKESID+4+Y,WV(X)-1:REM NULLO
                                   :rem 27
369 NEXT X:GOSUB60200              :rem 235
370 K$="AWSEDFTGYHUJKOLP:;*=↑"+CHR$(13)+"
    0":REM 0=PAUSA                 :rem 20
371 LK=LEN(K$):KK=2↑(1/12)         :rem 191
372 F1=523.251132:REM PRIMA NOTA:rem 15
373 DIM FR(LK),K(255)              :rem 91
374 FORI=1TOLK-1                   :rem 212
375 K=ASC(MID$(K$,I,1)):K(K)=I     :rem 134
376 FR(I)=5.8*F1+30                :rem 217
378 F1=F1*KK:NEXT I                :rem 79
379 FORX=0TO2:Y=7*X:PA(X)=SID+1+Y:PB(X)=S
    ID+Y:NEXT                      :rem 173
380 K(I)=0:FR(I)=0:REM PAUSA=NOTA '0'
                                   :rem 75
381 RETURN                         :rem 124
500 REM***************             :rem 25
501 REM*    MENU #2    *           :rem 88
502 REM***************             :rem 27
505 PRINT"{HOME}{ 14 GIU'}"SPC(S1)"
    { 6 SPAZI}PREMI {RVS}RETURN{OFF} PER
    RICOMINCIARE{GIU'}"            :rem 217
506 PRINTSPC(S1)"{RVS}1{OFF}-SUONARE
    { 12 SPAZI}{RVS}2{OFF}-AGG.NOTE
    { 2 SPAZI}"                    :rem 124
507 PRINTSPC(S1)"{GIU'}{RVS}3{OFF}-CORREG
    GERE{ 2 SPAZI}NOTE{ 3 SPAZI}{RVS}4
    {OFF}-SAVE SU DISCO"           :rem 166
```

```
508 PRINTSPC(S1)"{GIU'}{RVS}5{OFF}-SAVE S
    U NASTRO{ 5 SPAZI}{RVS}6{OFF}-CAMBIA
    INTROD"                          :rem  98
509 PRINTSPC(S1)"{GIU'}{RVS}7{OFF}-CAMBIA
     VOCE{ 8 SPAZI}{RVS}8{OFF}-OPZIONE DI
    SP."                             :rem 152
510 PRINTSPC(S1)"{ 2 SPAZI}[< 13 U>]{RVS}
     SCEGLI! {OFF}[< 13 U>]{HOME}":GOSUB6
    0200                             :rem  19
512 GETA$:IFA$<>""THEN512            :rem 144
513 GETA$:IFA$=""THEN513             :rem  85
514 GOSUB3000:IFA$=CHR$(13)THENGOTO2800
                                     :rem  48
515 IFVAL(A$)=0THEN505               :rem 249
516 ONVAL(A$)GOSUB1200,2900,5000,2600,160
    0,520,30000,31000                :rem 230
517 GOTO505                          :rem 112
520 PRINT"{CLR}{ 2 GIU'}ATTENDI MENTRE SA
    LVO LE NOTE..."                  :rem 199
530 SV=6*4096-2:I=0                  :rem   0
540 I=I+1:POKESV+2*I,N(I):POKESV+2*I+1,P(
    I):PRINT"{HOME}NOTA{ 3 SPAZI}
    { 3 SIN}"I                       :rem 126
550 IFI<MTHEN540                     :rem 195
555 FORJ=1TO20:POKESV+2*I+J,0:NEXT
                                     :rem 111
560 PRINT"{CLR}{ 6 GIU'}{WHT}STO CAMBIAND
    O I DATI{BLU}"                   :rem 104
595 POKE53281,6                      :rem  55
600 PRINT"{HOME}{ 3 GIU'}50009DATA"M:POKE
    198,3:POKE631,13:POKE632,13:POKE633,1
    3                                :rem  56
605 PRINT"K=50010:M="M":I=-19:SV=6*4096-1
    "                                :rem  71
610 PRINT"{ 2 GIU'}GOTO620{HOME}":END
                                     :rem 127
620 I=I+20:S$="":IFI>2*MTHENPRINT"{CLR}
    {BLK}":POKE53281,15:LIST50300-50305
                                     :rem 216
630 PRINT"{CLR}{ 3 GIU'}";:POKE198,3:POKE
    631,13:POKE632,13:POKE633,13:rem 145
632 PRINT"{WHT}"K"{BLU}DATA";        :rem 116
635 FORJ=0TO9:S=PEEK(SV+I+2*J):P=PEEK(SV+
    I+2*J+1)                         :rem 205
640 A$=MID$(STR$(S),2):B$=MID$(STR$(P),2)
                                     :rem 242
650 S$=S$+A$+","+B$+",":NEXT         :rem 150
660 PRINTS$+"{SIN} ":K=K+1:PRINT"K="K":M=
    "M":I="I":SV="SV                 :rem 162
665 PRINT"{ 2 GIU'}GOTO620{HOME}":END
                                     :rem 137
1000 REM*********************        :rem  23
1001 :                               :rem 252
1002 REM** CREA LA TASTIERA**        :rem  84
1003 :                               :rem 254
1004 REM*********************        :rem  27
1005 N=1                             :rem 130
1006 P1$="{OFF}{BLK}[<M>]":P2$=P1$:P3$=P1
     $                               :rem  59
1007 K1$="{RVS}{BLK} {WHT} {BLK} ":FORI=1
     TO12:P1$=P1$+K1$:NEXT           :rem   3
1008 P1$=P1$+"{RVS}{BLK} {WHT} {BLK}
     {OFF}"                          :rem 136
1010 K2$="{RVS}{WHT}{ 2 SPAZI}{OFF}{BLK}
     [<N>]":FORI=1TO13:P2$=P2$+K2$:NEXT:L
     =LEN(P2$):P2$=LEFT$(P2$,L-1)
                                     :rem 128
1020 P3$=P3$+"{RVS}{WHT}{ 2 SPAZI}
     { 6 DES}{OFF}{BLK}[<M>]{RVS}{WHT}
     { 10 DES}{OFF}{BLK}[<M>]{RVS}{WHT}
     { 7 DES}{OFF}{BLK}[<M>]{RVS}{WHT}
     { 8 DES}"                       :rem 247
1025 PRINT"{HOME}{GIU'}":FORI=1TO12:PRINT
     P2$:NEXT                        :rem 129
1030 PRINT"{HOME}{GIU'}":FORI=1TO5:PRINTP
     1$;:NEXT                        :rem 137
1035 PRINT"{HOME}{GIU'}":FORI=1TO5:PRINTP
     3$;                             :rem  23
1040 GOSUB200:N=N+5:FORPP=1TO100:NEXT
                                     :rem  38
1045 NEXT                            :rem   9
1050 PRINT"{HOME}{GIU'}{OFF}{BLK}
     [< 40 @>]"                      :rem 125
1055 PRINT"{HOME}{ 13 GIU'}";        :rem 199
1060 PRINT"{BLK}[< 40 T>]"           :rem 160
1070 GOSUB1300:RETURN                :rem  38
1100 REM*********************        :rem  66
1101 :                               :rem 253
1102 REM** LETT. TASTIERA    **      :rem  20
1103 :                               :rem 255
1104 REM*********************        :rem  70
1105 PRINT"{HOME}"SPC(S1)"{ 3 SPAZI}{RVS}
     W{OFF}[<G>] {RVS}E{OFF}{ 5 SPAZI}
     {RVS}T{OFF}{ 2 SPAZI}{RVS}Y{OFF}
     { 2 SPAZI}{RVS}U{OFF}{ 5 SPAZI}{RVS}
     O{OFF}{ 2 SPAZI}{RVS}P{OFF}
     { 5 SPAZI}{RVS}*{OFF}{ 2 SPAZI}{RVS}
     ↑"                              :rem  70
1110 PRINT"{ 13 GIU'}"SPC(S1);       :rem 103
1115 PRINT"{ 2 SPAZI}{RVS}A{OFF}
     { 2 SPAZI}{RVS}S{OFF}{ 2 SPAZI}{RVS}
     D{OFF}{ 2 SPAZI}{RVS}F{OFF}
     { 2 SPAZI}{RVS}G{OFF}{ 2 SPAZI}{RVS}
     H{OFF}{ 2 SPAZI}{RVS}J{OFF}
     { 2 SPAZI}{RVS}K{OFF}{ 2 SPAZI}{RVS}
     L{OFF}{ 2 SPAZI}{RVS}:{OFF}
     { 2 SPAZI}{RVS};{OFF}{ 2 SPAZI}{RVS}
     ={OFF} {RVS}RTN"                :rem 143
1120 RETURN                          :rem 164
1200 REM****************             :rem  71
1201 :                               :rem 254
1202 REM**  SUONA      **            :rem 215
1203 :                               :rem   0
1204 REM****************             :rem  75
1205 POKESID+24,15:ZZ=64             :rem 143
1206 FORQ=1TOM:N=N(Q):P=P(Q)         :rem 205
1209 AA=TI:SP=0:IFN>=23THEN1230      :rem  94
1210 SP=SP(N):H=H(N)                 :rem 140
```



```
1230 PRINT"{HOME}{ 4 SPAZI}{ 4 SIN}"Q
                                    :rem 111
1240 GOSUB200:POKEV3,0              :rem 39
1245 IFSP<>0THEN1270                :rem 161
1250 IFTI<AA+PTHEN1250              :rem 36
1255 GOTO1296                       :rem 216
1259 :                              :rem 11
1260 REM* SUONO ASSOCIATO AI TASTI *
                                    :rem 74
1270 X=SP>255:Y=SP>ZZ:POKEV0,-4*X:SP=SP+2
     55*X:SS=SP+ZZ*(SP>ZZ)          :rem 244
1275 POKEV1,SP:POKEV2,H:POKEV4,SS:POKEV5,
     H:POKEV3,4-8*Y                 :rem 32
1280 IFTI<AA+P THEN1280             :rem 42
1285 GOSUB60100                     :rem 71
1296 NEXT:FORPP=1TO1000:NEXT:POKEV3,0
                                    :rem 107
1297 GOSUB60200:RETURN              :rem 101
1300 REM*****************           :rem 114
1301 :                              :rem 255
1302 REM** SUONO PIANO **           :rem 93
1303 :                              :rem 1
1304 REM*****************           :rem 118
1305 POKESID,0:RETURN:PRINT@*!:POKESID+24
     ,15                            :rem 12
1310 FORPP=1TO10:R=RND(1):N=23*R+1:GOSUB2
     00                             :rem 59
1315 POKEV1,255*R:POKEV2,100+30*R:POKEV3,
     4:FORQQ=1TO10:NEXT:NEXT:POKEV3,0
                                    :rem 67
1320 GOSUB60100:FORPP=0TO500:NEXT
                                    :rem 28
1325 GOSUB60200:RETURN              :rem 93
1400 REM*****************           :rem 115
1401 :                              :rem 0
1402 REM** INPUT NOTE  **           :rem 25
1403 :                              :rem 2
1404 REM*****************           :rem 119
1405 M=0:N=0:GOTO1460               :rem 183
1410 M=M+1:M$=MID$(STR$(M),2):PRINT"
     {HOME}{ 3 SPAZI}{ 3 SIN}"M$:rem 175
1411 GETN$:IFN$=""THENGOSUB60100:GOTO1411
                                    :rem 181
1412 NN=ASC(N$+Z$):AA=TI            :rem 122
1415 IFNN=N0THENM=M-1:GOSUB3000:RETURN
                                    :rem 172
1420 IFNN=N1THENM=M-1:GOTO1411      :rem 76
1421 M$=MID$(STR$(M),2)             :rem 222
1422 PRINT"{HOME}{ 3 SPAZI}{ 3 SIN}"M$
                                    :rem 245
1425 N=K(NN)                        :rem 143
1430 GOSUB200:SP=SP(N):H=H(N)       :rem 220
1435 IFSP=0THEN1445                 :rem 105
1438 X=SP>255:POKEV0,-4*X:SP=SP+255*X
                                    :rem 230
1440 POKEV1,SP:POKEV2,H:POKEV3,4:rem 5
1445 IFPEEK(KB)<>64THEN1445:REM ATTENDE
                                    :rem 100
1446 P=TI-AA                        :rem 168
1450 GOSUB60100                     :rem 65
1456 POKEV3,0                       :rem 228
1457 GETA$:IFA$<>""THEN1457         :rem 2
1460 N(M)=N:P(M)=P:IFM>499THENGOSUB3000:R
     ETURN                          :rem 125
1465 GOTO1410                       :rem 207
1600 REM************************
                                    :rem 197
1601 :                              :rem 2
1602 REM** SALVA SU NASTRO        **
                                    :rem 75
1603 :                              :rem 4
1604 REM************************
                                    :rem 201
1612 GOSUB3000                      :rem 13
1620 PRINTSD$"{GIU'}NOME DEL FILE
     { 3 SPAZI}";                   :rem 197
1625 INPUTNM$                       :rem 29
1650 OPEN1,1,1,NM$                  :rem 212
1655 PRINT#1,M:PRINT#1,Z$           :rem 240
1660 FORI=1TOM                      :rem 91
1665 PRINT#1,N(I)                   :rem 199
1666 PRINT#1,P(I)                   :rem 202
1670 NEXT:CLOSE1                    :rem 238
1675 GOSUB3000                      :rem 22
1685 RETURN                         :rem 180
1700 REM************************
                                    :rem 198
1701 :                              :rem 3
1702 REM** CARICA DA NASTRO       **
                                    :rem 85
1703 :                              :rem 5
1704 REM************************
                                    :rem 202
1705 PRINTSD$                       :rem 21
1715 INPUT"NOME DEL FILE :{ 3 SPAZI}V
     { 3 SIN}";NM$                  :rem 167
1720 IFNM$="V"THEN1705              :rem 107
1725 GOSUB2000                      :rem 17
1730 GOSUB60100                     :rem 66
1740 OPEN1,1,0,NM$                  :rem 211
1745 INPUT#1,M:INPUT#1,Z$           :rem 246
1760 FORI=1TOM                      :rem 92
1765 INPUT#1,N(I)                   :rem 203
1766 INPUT#1,P(I)                   :rem 206
1775 NEXT                           :rem 19
1780 CLOSE1                         :rem 119
1782 GOSUB3000:PRINTSD$+"{RVS}"+NM$
                                    :rem 2
1785 GOSUB1200                      :rem 24
1795 RETURN                         :rem 182
1800 REM***************************
                                    :rem 69
1801 :                              :rem 4
1802 REM** INPUT NOTA CORRETTA     **
                                    :rem 125
1803 :                              :rem 6
1804 REM***************************
                                    :rem 73
1805 GETN$                          :rem 32
1810 AA=TI:GETN$:SN=PEEK(KB):IFN$=""THEN1
     810                            :rem 130
1815 NN=ASC(N$):N=K(NN)             :rem 63
1820 GOSUB200                       :rem 221
1830 REM* ATTENDE LA MODIFICA * :rem 210
1835 IFPEEK(KB)=SNTHEN1835          :rem 65
1840 P=INT((TI-AA)/2+1)             :rem 240
1845 GOSUB60200:RETURN              :rem 100
1900 REM***********                 :rem 124
1901 :                              :rem 5
1902 REM** TITOLO**                 :rem 51
1903 :                              :rem 7
1904 REM***********                 :rem 128
1905 PRINTSD$"{BLK}"                :rem 235
1910 PRINTSPC(12)"{RVS}{ 2 SPAZI}[<*>]
     {OFF} {RVS} {OFF} {RVS}{ 3 SPAZI}
     {OFF} {RVS} {OFF} {RVS} {OFF} {RVS}
     { 3 SPAZI}{OFF}"               :rem 237
1915 PRINTSPC(12)"{RVS} {OFF} {RVS} {OFF}
     {RVS} {OFF} {RVS} {OFF} {RVS} {OFF}
     {RVS} {OFF} {RVS} {OFF} {RVS} {OFF}
     {RVS} {OFF}"                   :rem 255
```

```
1920 PRINTSPC(12)"{RVS} {OFF} {RVS} {OFF}
     {RVS} {OFF} {RVS}{ 3 SPAZI}{OFF}
     {RVS} [<*>] {OFF} {RVS} {OFF} {RVS}
     {OFF}"                    :rem 146
1925 PRINTSPC(12)"{RVS}{ 2 SPAZI}{OFF}
     £ {RVS} {OFF} {RVS} {OFF} {RVS}
     {OFF} {RVS} {OFF}[<*>]{RVS} {OFF}
     {RVS} {OFF} {RVS} {OFF}"   :rem 228
1930 PRINTSPC(12)"{RVS} {OFF}{ 3 SPAZI}
     {RVS} {OFF} {RVS} {OFF} {RVS} {OFF}
     {RVS} {OFF} {RVS} {OFF} {RVS}
     { 3 SPAZI}"                :rem 34
1940 FORPP=0TO19:POKE1984+PP,102:POKE2023
     -PP,102:NEXT              :rem 205
1950 FORPP=0TO19:POKE56256+PP,6:POKE56295
     -PP,6:NEXT                :rem 138
1960 FORPP=0TO19:POKE1024+PP,102:POKE1063
     -PP,102:NEXT              :rem 195
1970 FORPP=0TO19:POKE55296+PP,6:POKE55335
     -PP,6:NEXT                :rem 137
1990 RETURN                    :rem 179
2000 REM*****************      :rem 154
2001 :                         :rem 253
2002 REM** CANCELLA       *    :rem 89
2003 :                         :rem 255
2004 REM*****************      :rem 158
2010 FORPP=0TO39:POKECRT+PP,32:NEXT
                               :rem 237
2020 FORPP=0TO39:POKECRT+14*WD+PP,32:NEXT
                               :rem 67
2030 RETURN                    :rem 165
2100 REM***********************:rem 151
2101 :                         :rem 254
2102 REM**  TASTIERA/NO SUONO **:rem 14
2103 :                         :rem 0
2104 REM***********************:rem 155
2106 FORJ=1TO6:PRINT"{HOME}":FORK=1TOJ:PR
     INT"{GIU'}";:NEXT:PRINTSPC(S1);
                               :rem 179
2110 FORI=1TO13:PRINT" {RVS} {OFF} ";:NEX
     T:NEXT                    :rem 69
2115 FORJ=1TO6:PRINT"{HOME}{ 5 GIU'}":FOR
     K=1TOJ:PRINT"{GIU'}";:NEXT:PRINTSPC(
     S1);                      :rem 8
2120 FORI=1TO13:PRINT"{RVS}[<G>] [<M>]";:
     NEXT:NEXT                 :rem 0
2125 FORJ=1TO6:PRINT"{HOME}":FORK=1TOJ:PR
     INT"{GIU'}";:NEXT         :rem 247
2130 PRINTSPC(S1)"{RVS}[<G>] "SPC(6)"
     {RVS}[<M>][<G>]"SPC(10)"{RVS}[<M>]
     [<G>]"SPC(7)"{RVS}[<M>][<G>]":NEXT
                               :rem 219
2135 PRINT"{HOME}{GIU'}"SPC(S1)"
     [< 39 @>]"                :rem 187
2140 PRINT"{HOME}{ 13 GIU'}"SPC(S1);
                               :rem 126
2145 PRINT"[< 39 Y>]"          :rem 126
2150 RETURN                    :rem 168
2200 REM*****************      :rem 114
2201 :                         :rem 255
2202 REM*INSERIM. NOTE   *     :rem 121
2203 :                         :rem 1
2204 REM*****************      :rem 118
2205 PRINT"{HOME}{ 16 GIU'}":C$="{OFF}"
                               :rem 113
2210 GOSUB60100:FORL=1TO10     :rem 27
2215 PRINTSPC(S1)C$"SUONA!!{ 7 SPAZI}"
                               :rem 133
2216 PRINT"{SU}";              :rem 104
2220 IFC$="{OFF}"THENC$="{RVS}":GOTO2230
                               :rem 50
2225 C$="{OFF}"                :rem 69
2230 FORK=1TO100:NEXT:NEXT     :rem 141
2235 PRINT:RETURN              :rem 115
2500 REM*****************      :rem 159
2501 :                         :rem 2
2502 REM** ISTRUZIONI    **    :rem 117
2503 :                         :rem 4
2504 REM*****************      :rem 163
2505 PRINT"{HOME}{ 16 GIU'}"   :rem 192
2510 PRINTSPC(S1)"{RVS}{ 15 SPAZI}TERMINA
     CON{ 2 SPAZI}UN "CHR$(34)"."CHR$(34
     )"{ 3 SPAZI}"             :rem 33
2515 PRINTSPC(S1)"{GIU'}{RVS} INSERISCI U
     NO "CHR$(34)"0"CHR$(34)" PER UNA PAU
     SA{ 6 SPAZI}"             :rem 134
2520 PRINTSPC(S1)"{GIU'}{RVS} DOPO UNA NO
     TA ERRATA BATTI "CHR$(34)"-"CHR$(34)
     "{ 6 SPAZI}"              :rem 26
2525 RETURN                    :rem 174
2600 REM***********************
                               :rem 198
2601 :                         :rem 3
2602 REM** SALVA SU DISCO     **
                               :rem 231
2603 :                         :rem 5
2604 REM***********************
                               :rem 202
2612 GOSUB3000:NW=WV(0)        :rem 88
2620 PRINTSD$"{GIU'}NOME DEL FILE
     { 3 SPAZI}";              :rem 198
2625 INPUTNM$:IFLEFT$(NM$,1)<>" "THENNM$=
     " "+NM$                   :rem 186
2626 GOSUB3000                 :rem 19
2635 DR$="0":D0$=DR$           :rem 4
2636 GOSUB3000                 :rem 20
2637 PRINTSD$"{GIU'}SOSTITUISCO FILE ESIS
     TENTE";:INPUT"{ 3 SPAZI}N{ 3 SIN}";A
     $                         :rem 186
2638 IFLEFT$(A$,1)="S"THEND0$="@"+DR$
                               :rem 5
2640 OPEN15,8,15:R$=CHR$(13)   :rem 253
2650 OPEN2,8,2,D0$+":"+NM$+",S,W"
                               :rem 187
2651 GOSUB40000:IFE0<>0THEN2670 :rem 40
2655 PRINT#2,M;R$;NW;R$        :rem 111
2660 FORI=1TOM                 :rem 92
2665 PRINT#2,N(I);R$;P(I);R$:NEXT
                               :rem 201
2670 CLOSE2:CLOSE15            :rem 141
2675 GOSUB3000:RETURN          :rem 49
2700 REM***********************
                               :rem 199
2701 :                         :rem 4
2702 REM** CARICA DA DISCO    **
                               :rem 241
2703 :                         :rem 6
2704 REM***********************
                               :rem 203
2705 GOSUB63900:GOSUB3000      :rem 205
2710 PRINTSD$;                 :rem 77
2715 INPUT"NOME DEL FILE :{ 3 SPAZI}V
     { 3 SIN}";NM$:IFNM$="V"THEN2710
                               :rem 128
2720 IFLEFT$(NM$,1)<>" "THENNM$=" "+NM$
                               :rem 45
2730 GOSUB3000:OPEN15,8,15:PRINTSD$;
                               :rem 148
2740 OPEN2,8,2,DR$+":"+NM$+",S,R"
                               :rem 216
2745 GOSUB40000:IFE0<>0THENCLOSE2:CLOSE15
```

```
     :RETURN                              :rem  53
2750 INPUT#2,M,NW                         :rem 253
2760 FORI=1TOM                            :rem  93
2765 INPUT#2,A$,B$                        :rem 220
2770 N(I)=VAL(A$):P(I)=VAL(B$):NEXT
                                          :rem   2
2775 CLOSE2:CLOSE15:GOSUB30050            :rem  69
2776 GOSUB3000:PRINTSD$+"{ 3 GIU'}
     { 3 DES}{RVS}"+NM$                   :rem 144
2790 GOSUB1200:GOSUB3000:RETURN           :rem 172
2800 REM***************                   :rem  36
2801 REM*   MENU #1    *                  :rem 140
2802 REM***************                   :rem  38
2804 GOSUB3000:GOSUB2000:GOSUB60200
                                          :rem  63
2805 PRINTSD$"{RVS}1{OFF}:CARICA DA DISCO
     "                                    :rem   4
2810 PRINTSPC(S1)"{GIU'}{RVS}2{OFF}:CARIC
     A DA NASTRO"                         :rem 119
2815 PRINTSPC(S1)"{GIU'}{RVS}3{OFF}:COMPO
     NI MUSICA"                           :rem  85
2816 PRINTSPC(S1)"{GIU'}{RVS}4{OFF}:FINE
     {GIU'}"                              :rem 179
2820 FORI=1TO3                            :rem  64
2825 PRINT"{SU}{ 39 SPAZI}":FORPP=1TO100:
     NEXT                                 :rem  15
2830 PRINT"{SU}{ 2 SPAZI}[< 13 U>]{RVS} S
     CEGLI! {OFF}[< 12 U>]":FORPP=1TO100:
     NEXT                                 :rem 127
2835 NEXT                                 :rem  17
2840 GOSUB60200                           :rem  70
2845 GETA$:IFA$=""THEN2845                :rem 201
2850 GOSUB3000                            :rem  18
2855 IFVAL(A$)=0THEN2805                  :rem 103
2860 ONVAL(A$)GOTO2865,2870,80            :rem 165
2861 PRINTSD$:POKESID+24,0:END            :rem  96
2865 GOSUB2700:GOTO500                    :rem  38
2870 GOSUB1700:GOTO500                    :rem  33
2900 REM*************                     :rem 209
2901 :                                    :rem   6
2902 REM* AGG.NOTE   *                    :rem  56
2903 :                                    :rem   8
2904 REM*************                     :rem 213
2905 :                                    :rem  10
2925 GOSUB3000:GOSUB1100                  :rem 145
2940 GOSUB2500:GOSUB2200                  :rem 148
2950 GOSUB1410:GOSUB2000                  :rem 146
2960 GOSUB1200:RETURN                     :rem  46
3000 REM**************                    :rem 243
3001 :                                    :rem 254
3002 REM**CANCELLA   **                   :rem 132
3003 :                                    :rem   0
3004 REM**************                    :rem 247
3005 PRINT"{HOME}{ 15 GIU'}";             :rem 230
3010 FORT=1TO9:PRINTCL$:NEXT              :rem  60
3020 GOSUB2000:IFQW=1THENRETURN           :rem 245
3040 PRINT"{HOME}";:RETURN                :rem   0
5000 REM**********************            :rem  69
5001 :                                    :rem   0
5002 REM**CANC.NOTE ERRATE   **           :rem 143
5003 :                                    :rem   2
5004 REM**********************            :rem  73
5016 GOSUB3000                            :rem  15
5025 PRINTSD$"{GIU'} PREMI {RVS}@{OFF} QU
     ANDO SUONO LA NOTA ERRATA. "
                                          :rem 210
5026 PRINTSPC(S1)"{GIU'}CI SONO "M" NOTE.
     .."                                  :rem 134
5027 PRINTSPC(S1)"{GIU'}INIZIO CON NOTA #
     ";:INPUT"{ 3 SPAZI}[<+>]{ 3 SIN}";PZ
                                          :rem  29
5028 PRINTSPC(S1)"{GIU'}VEL.VISUALIZZAZ.(
     1-10:1=VELOCE)";:INPUT"{ 3 SPAZI}
     [<+>]{ 3 SIN}";PY                    :rem  59
5030 FORQ=PZTOM:N=N(Q):P=P(Q)             :rem  69
5031 AA=TI:SP=0:IFN>=23THEN5033           :rem  96
5032 SP=SP(N):H=H(N)                      :rem 146
5033 PRINT"{HOME}{ 4 SPAZI}{ 4 SIN}"Q
                                          :rem 116
5034 GOSUB200                             :rem 222
5035 IFSP<>0THEN5038                      :rem 168
5036 IFTI<AA+PTHEN5036                    :rem  48
5037 GOTO5044                             :rem 213
5038 X=SP>255:POKEV0,-4*X:SP=SP+255*X
                                          :rem 230
5039 POKEV1,SP:POKEV2,H:POKEV3,4:rem       13
5040 IFTI<AA+P THEN5040                   :rem  38
5041 GOSUB60100                           :rem  65
5042 POKEV3,0                             :rem 223
5044 FORK=1TO10*PY                        :rem  68
5045 GETN$:BN=ASC(N$+Z$)                  :rem 106
5050 IFBN=N2THENK=100:SQ=Q:Q=M            :rem 111
5055 NEXT:NEXT                            :rem 135
5060 GOSUB3000                            :rem  14
5065 GOSUB1100                            :rem  18
5070 PRINTSD$"{GIU'} INSERISCI NOTA CORRE
     TTA #"SQ                             :rem 111
5075 GOSUB1800                            :rem  26
5080 N(SQ)=N::P(SQ)=P:GOSUB3000           :rem  94
5085 GOSUB1200                            :rem  21
5090 GOSUB60100:GOSUB2000:RETURN:rem       219
10000 REM******************               :rem 243
10001 :                                   :rem  44
10002 REM**  INIZIALIZZA   **             :rem 205
10003 :                                   :rem  46
10004 REM******************               :rem 247
10010 GOSUB1900:Z$=CHR$(0)                :rem 179
10020 GOSUB350:PS=ASC("0")                :rem 193
10025 KB=203                              :rem  87
10030 SD$="{HOME}{ 15 GIU'}"              :rem  66
10040 N0=ASC("."):N1=ASC("-"):N2=ASC("@")
      :WD=40:CRT=1024                     :rem  25
10050 VD=53248                            :rem 211
10060 V0=VD+16:V1=VD+4:V2=VD+5:V3=VD+21
                                          :rem 254
10065 V4=VD+6:V5=VD+7                     :rem 188
19000 D(0)=55295+10*WD:D(1)=55295+5*WD
                                          :rem 195
19010 D3=CRT+14*WD:D4=D3+WD-1             :rem 119
19060 CR(1)=1:CR(0)=0                     :rem  66
20000 REM*****************************
                                          :rem 152
20001 :                                   :rem  45
20002 REM**NOTE DA TASTIERA          **
                                          :rem 152
20003 :                                   :rem  47
20004 REM*****************************
                                          :rem 156
20005 DIMSP(255),H(255)                   :rem 194
20010 FORI=1TOLK                          :rem 203
20020 READSP,H:SP(I)=8*SP:H(I)=130-30*H:N
      EXT:SP(0)=0                         :rem  19
20030 GOSUB60200                          :rem 109
20050 DATA3,0,4,1,6,0,7,1,9,0,12,0,13,1,1
      5,0,16,1,18,0,19,1,21,0             :rem  33
20055 DATA24,0,25,1,27,0,28,1,30,0,33,0,3
      4,1,36,0,37,1,39,0,0,0              :rem   5
20060 PRINTSPC(11)"{GIU'}SUONO L'INTRODUZ
      IONE?"                              :rem  99
20062 CL$="{ 39 SPAZI}"                   :rem  46
```

```
20063 BR$="{RED}{RVS}{ 40 SPAZI}{OFF}
      {BLK}"                          :rem  132
20065 GETA$:IFA$=""THEN20065          :rem   29
20066 PRINT"{CLR}"                    :rem   98
20067 IFA$<>"S"THENNO=1               :rem   62
20070 GOSUB32000                      :rem  110
20100 RETURN                          :rem  211
30000 REM CAMBIA VOCI                 :rem  165
30001 REM"[< 13 U>]                   :rem   82
30010 IFWV(0)=17THENNW=33             :rem  205
30020 IFWV(0)=33THENNW=65             :rem  209
30030 IFWV(0)=65THENNW=17             :rem  212
30040 PRINTSD$"{ 12 SPAZI}{RVS}F. ONDA "N
      W:FORPP=1TO500:NEXT             :rem  176
30050 IFNW=0THENNW=17                 :rem   18
30060 FORPP=0TO2:WV(PP)=NW:NEXT       :rem  245
31000 REM OPZIONE LIBERA              :rem  171
31001 REM"[< 14 U>]                   :rem   11
31010 RETURN                          :rem  213
32000 REM MANI                        :rem  254
32001 REM"[< 5 U>]                    :rem  148
32002 FORPP=0TO62:READX:POKE2048+PP,X:NEX
      T                               :rem  163
32003 FORPP=0TO62:READX:POKE2112+PP,X:NEX
      T                               :rem  156
32004 POKE2042,32:POKE2043,33:POKEVD+21,0
                                      :rem   50
32005 POKEVD+41,09:POKEVD+42,09:REM COLOR
      E                               :rem  243
32006 POKEVD+23,12:POKEVD+29,12:REM ESPAN
      DE                              :rem   41
32007 POKEVD+27,0:REM PRIORITA'       :rem  100
32009 RETURN                          :rem  222
32010 DATA1,128,0,2,64,0,3,193,128,3,194,
      64                              :rem   98
32015 DATA3,195,192,3,199,156,3,199,60,22
      7,207,123                       :rem  223
32020 DATA243,222,247,123,221,238,63,255,
      254                             :rem  165
32025 DATA31,255,248,15,255,240,15,255,22
      4,7,255                         :rem  113
32030 DATA224,7,255,224,7,255,224,3,255,2
      24                              :rem  115
32035 DATA1,255,192,0,128,64,0,255,192
                                      :rem  125
32050 DATA0,1,128,0,2,64,1,131,192,2,67,1
      92                              :rem   93
32055 DATA27,227,192,61,227,192,222,243,1
      99,239                          :rem   83
32060 DATA115,207,247,251,222,125,255,252
                                      :rem   17
32065 DATA63,255,248,31,255,248,15,255,24
      8,7,255,248                     :rem   80
32070 DATA3,255,248,3,255,248,3,255,240,3
      ,255                            :rem  220
32085 DATA224,1,255,192,0,192,64,0,255,19
      2                               :rem   71
40000 REM ERRORI DISCO                :rem   29
40001 REM"[< 12 U>]                   :rem  155
40010 INPUT#15,E0,E1$,E2,E3$          :rem  224
40020 IFE0=0THENRETURN                :rem  118
40030 GOSUB3000:PRINTSD$"{ 2 GIU'}"E1$
                                      :rem  188
40040 FORPP=1TO1000:NEXT:GOSUB3000:RETURN
                                      :rem   97
50000 REM*********************        :rem  159
50001 :                               :rem   48
50002 REM**MUSICA INTRODUTTIVA **
                                      :rem  242
50003 :                               :rem   50
50004 REM*********************
                                      :rem  205
50005 READM:FORI=1TOM                 :rem   43
50006 READN(I),P(I):NEXT:GOSUB50300:GOTO1
      200                             :rem  118
50009 DATA 103                        :rem  172
50010 DATA12,2,8,2,10,2,12,2,15,2,13,2,13
      ,2,17,2,15,2,15,2               :rem    9
50011 DATA20,2,19,2,20,2,15,2,12,2,8,2,10
      ,2,12,2,13,2,15,2               :rem    6
50012 DATA17,2,15,2,13,2,12,2,10,2,12,2,8
      ,2,7,2,8,2,10,2                 :rem  171
50013 DATA3,2,7,2,10,2,13,2,12,2,10,2,12,
      2,8,2,10,2,12,2                 :rem  157
50014 DATA15,2,13,2,13,2,17,2,15,2,15,2,2
      0,2,19,2,20,2,15,2              :rem   66
50015 DATA12,2,8,2,10,2,12,2,5,2,15,2,13,
      2,12,2,10,2,8,2                 :rem  167
50016 DATA3,2,8,2,7,2,8,2,12,2,15,2,20,2,
      15,2,12,2,8,2                   :rem   84
50017 DATA12,2,15,2,18,2,15,2,12,2,8,2,12
      ,2,15,2,17,2,13,2               :rem   23
50018 DATA10,2,7,2,10,2,13,2,15,2,12,2,8,
      2,5,2,8,2,12,2                  :rem  126
50019 DATA13,2,10,2,7,2,3,2,7,2,10,2,13,2
      ,12,2,10,2,8,2                  :rem  120
50020 DATA12,2,15,2,20,0,0,0,0,0,0,0,0,0,
      0,0,0,0,0,0                     :rem  180
50300 PRINT"{HOME}{ 22 GIU'}"SPC(12)"****
      { 2 SPAZI}J O Y{ 3 SPAZI}****"
                                      :rem  114
50301 REM MODIFICA QUI SOPRA IL NOME DEL
      BRANO E PREMI 'RETURN', QUINDI SALV
      A                               :rem  245
50400 RETURN                          :rem  217
60000 REM TUTTE LE VOCI               :rem   50
60001 REM"[< 18 U>]                   :rem  237
60005 FORVV=0TO2:POKESID++4+7*VV,WV(VV)-1
      :NEXT:VV=0                      :rem  219
60010 FORPP=1TO80*RE:NEXT:RETURN:rem   75
60100 REM VOCE PRECEDENTE             :rem  231
60101 REM"[< 22 U>]                   :rem  206
60110 POKESID+4+7*PV,WV(PV)-1         :rem  198
60130 RETURN                          :rem  218
60200 REM STOP SID                    :rem    2
60201 REM"[< 8 U>]                    :rem  191
60210 FORPP=OTO2:POKESID+4+7*PP,0:NEXT:VV
      =0:RETURN                       :rem  247
63900 GOSUB3000:PRINTSD$;:DR$="0":GOSUB30
      00:B=30:AN=1                    :rem  119
63901 PRINTSD$"BRANI PRESENTI SUL DISCO:"
                                      :rem    5
63902 OPEN15,8,15                     :rem  146
63904 OPEN2,8,0,"$"+DR$+":*=S"        :rem    6
63905 GET#2,A$:GET#2,A$:I$=""         :rem  245
63906 GETB$:IFST<>0ORB$<>""THEN63918
                                      :rem  103
63907 FORA=1TOB:GET#2,A$              :rem  132
63908 IFA$=""THENA$=CHR$(0)           :rem  210
63909 I$=I$+A$:NEXT:IFAN=1THEN63915
                                      :rem   40
63910 PRINTSD$"{ 2 GIU'}{ 37 SPAZI}":PP=7
                                      :rem  255
63911 U1$=MID$(I$,PP,19):IFLEFT$(U1$,1)<>
      CHR$(34)THENPP=PP+1:IFPP<19THEN6391
      1                               :rem  246
63912 IFPP>19THEN63916                :rem   20
63914 IFMID$(U1$,2,1)=" "THENPRINTSD$"
      { 2 GIU'}"U1$                   :rem   82
63915 FORI1=1TO10:I2=TAN(I1):NEXT
                                      :rem  244
63916 AN=0:I$="":B=32:GOTO63906       :rem  198
63918 CLOSE2:CLOSE15:RETURN           :rem  227
```

input·output

Piccoli annunci



Vendo super expander per VIC 20: aggiunge 3 Kbyte RAM e 15 nuovi comandi. Cambio software per VIC inespanso. Stefano Succi - Via Matteotti, 41 - 47030 Borghi (BO) - Tel. 0451/939077

Vendo e cambio tantissimi splendidi programmi (circa 400) su disco e nastro. Prezzi stracciatissimi. Cerco commodoristi (C64) per scambio liste e software. Roberto Pallotti - Via Audinot, 35 - 40134 Bologna - Tel. 051/42195

Vendo VIC 20 come nuovo, completo di cavetti ed alimentatore, registratore dedicato con numerosissimi programmi su cassette e su listati, libro 46 programmi vari. Prezzo da concordare. Sergio Casciu - Via Domodossola, 21 - 20145 Milano - Tel. 3491122

Vendo giochi, utility, linguaggi (Simons') di altissimo livello per C64 a prezzi concorrenziali. Permuto con FORTH e Pascal. Telefonare dopo le 20.00. Michele Persiani - Via Villa Meloise, 9 - 90146 Palermo - Tel. 253801

Vendo per CBM 64 giochi ed utility su cassetta a L. 1.500 cadauno. Per richiedere la lista telefonare dalle 14 alle 15 a: Settimio Capuccini - Via Firenze, 2 - 67059 Trasacco (AQ) - Tel. 0863/93237

Vendo e cambio giochi e programmi di utilità sprotetti per CBM 64. Se interessati richiedere lista. Sono disposto a cedere più di 100 programmi per un plotter per C64. Nicola Rosatelli - Via dei Limoni, 27 - 00172 Roma - Tel. 06/282864

Vendo programmi per C64 in linguaggio macchina a solo L. 4.000 cadauno. Telefonare a: Pasquale Del Pezzo - Via F.S. Caiazzo, 29 - 84012 Angri (SA)

Vendo bellissimi giochi per C64: 50 giochi su cassetta con Turbo Tape a L. 8.000. Telefonare nel pomeriggio di lunedì - mercoledì - giovedì. Stefano Bologna - Via Scheiwiller, 1 - 20139 Milano - Tel. 02/562231

Vendo i più bei giochi per C64: Pole Position (Atari), Dig Dug (Atari), Decathlon (Activision), Pitfall II (Activision), Popeye (Parker), Gyruss (Parker), Frogger, Zaxxon e tanti altri entusiasmanti videogames. Giuseppe Mazzanti - Via San Salvatore, 29 - 72024 Oria (BR) - Tel. 0831/945247

Vendo e cambio programmi su disco e cassetta per C64. Sono in possesso di giochi parlanti, word processor professionali, tante utility e giochi per tutti i gusti. Invio liste a tutti. Vendo in tutta Italia. Antongiulio Pisani - Via Posillipo, 69/9 - 80123 Napoli - Tel. 081/7695397

Vendo per VIC 20 4 cartucce: Gorf L. 40.000, Cosmic Cruncher L. 40.000, Mole Attack L. 40.000, Sargon II Chess L. 40.000. In blocco L. 150.000. Telefonare dalle 14.30 alle 15.30 Fabio Ferazza - Via Roma, 12 - 28050 Pombia (NO) - Tel. 0321/956248

Svendo (causa passaggio a sistema superiore) tutti i miei videogames per CBM 64, interamente in LM e di ottima qualità all'eccezionale prezzo di L. 1.000 cadauno. Inoltre posseggo giochi originali Commodore 64 (L. 5.000). Acquisto minimo di L. 20.000, in regalo Turbo Tape. Per informazioni rivolgersi a: Giacomo Brighina - Via Bouganvillea, 8 - 95041 Caltagirone (CT) - Tel. 0933/25877 991342

Sono interessato a scambi, acquisti e vendite di programmi per C64. Scrivete o telefonate a: Alessandro Cumin - Via Berlam, 3 - 34136 Trieste - Tel. 040/43280

Vendo VIC 20 più registratore, 4 utility, 55 giochi favolosi, manuale in italiano, joystick e listati vari. Il tutto in ottime condizioni a L. 240.000. Telefonare al pomeriggio. Enea Invernizzi - Via Matteotti, 5 - Cassina de Pecchi (MI) - Tel. 02/9528106

Solo cambio (non vendo) di giochi ed utility per C64; lista di oltre 200 titoli su cassetta in Turbo Tape. Scrivere a: Renzo Sartini - Via Pisana, 46 - 50018 Scandicci (FI) - Tel. 754601

Cambio plotter originale Commodore 1520 completo di penne, rotoli carta, tools, con stampante MPS801 o 803. Disposto anche a conguaglio. Giovanni Perrone - Via Don Lorenzo Milani, 7 - 50038 Scarperia (FI)

Regalo 100 giochi per VIC 20 a chi mi procura i numeri arretrati dall'1 al 6 della rivista Computer Club. Luciano Lucchetti - Via Oberdan, 8 - 17027 Pietra Ligure (SV) - Tel. 019/544024

Cambio programmi su disco (Missione Impossibile, Raid Over Moscow, ecc.). Circa 1.000 programmi. Cerco utilizzatori di modem 1200 baud. Flavio Bernardotti - Via Trento, 10 - 15040 Montecastello (AL)

Per C64 possiedo circa 150 giochi tra cui: Pole Position, Decathlon e il favoloso Turbo Tape e tantissimi altri. Scrivete o telefonate a: Emauele Silvestri - Piazza Malatesta, 2 - 47040 Verucchio (FO) - Tel. 0541/668040

Compro, vendo, cambio software per C64. I migliori programmi a sole L. 1.000 cadauno. Sconto per quantità. Dispongo di Turbo, Calcio, Basket, Tarzan, F1, Pac Man, ecc. Vincenzo Petrone - Via Umberto I, 26 - 85010 Anzi (PZ) - Tel. 0971/962019

Cambio per VIC 20 50 giochi su cassetta. Cambio solo in provincia di Alessandria. Massimo Boffitto - Via Donna - 15062 Bosco Marengo (AL) - Tel. 759398

Vendo VIC 20 perfetto più registratore, espansione da 3/8/16 Kbyte, 50 giochi vari, 2 cartridge, 1 libro di listati, 1 libro di BASIC. Romeo Berlato - Via G. Marconi, 83 - 36035 Marano (VI) - Tel. 0445/622002

Cambio programmi per C64. Inviate le vostre liste a: Nunzio Peligra - Via G. Cantore, 24 - 97013 Comiso (RG) - Tel. 965595

Vendo VIC 20 più paddles, joystick, 3 cassette gioco a L. 20.000 l'una, il tutto a L. 225.000. Riccardo Fulcali - Via Roma, 30 - 83040 Gesualdo (AV) - Tel. 401028

Vendo software. Sergio Biasetti - Via delle Terme, 23 - 80077 Ischia Porto (NA)

Occasionissima! Vendesi VIC 20 più registratore, joystick e molti giochi ed utility su cassetta a L. 200.000. Prego gli interessati di scrivere o telefonare a: Marcello Lo Jacono - Via Luigi Fantoni, 88 - 37069 Villafranca (VR) - Tel. 045/793491

Attenzione: il team CBM 64 vende, scambia programmi e cerca soci. Per informazioni telefonare o scrivere a: Luigi Prinetti - Via Di Vittorio, 7 - 29100 Piacenza - Tel. 0523/60408

Vendo super videogiochi e programmi di utility a prezzi da favola. Chiedetemi la lista e vedrete! Fate presto. Luca Gamberini - Via Tellera, 8 - 40133 Bologna - Tel. 051/569958

Eccezionale! Vendo per il CBM 64 25 videogiochi su cassetta a L. 30.000. Sono i migliori sul mercato: Calcio, Basket, Super Pipeline, One On One, Spy, Decathlon, Le Mans, Atzec, Squish'em, ecc. Telefonare ore pasti. Giuseppe De Vivo - Via Napoli, 167 - 71016 San Severo (FG) - Tel. 0882/73911

Vendo, cambio software per VIC 20. Inoltre vendo cartuccia "Adventureland" ad un prezzo speciale: solo L. 25.000. Stefano Cumini - Via Albisetti, 22 - 21049 Tradate (VA) - Tel. 845754

Conoscerei in Bologna utenti di C64 principianti e non per imparare insieme l'uso del computer e i programmi. Lorenzo Reggiani - Via Bianconi, 2 - 40137 Bologna - Tel. 051/342665

Compro, vendo, cambio software per C64 su disco con manuali: Easy Script, Simons' BASIC, Stat 64, Ingegneria Capurso, Sinford, The Manager, ecc. originali. Dispongo di altri programmi senza manuali. Regalo software giochi in cambio di manuali. Inviare l'elenco dei programmi, rispondo a tutti. Alessandro Scillitani - Via Candelaro, 28/2 - 71100 Foggia - Tel. 0881/40514

Vendo o cambio giochi per VIC 20 inespanso. Cassetta (a mio carico) con 5 giochi L. 6.000. Roberto Paolini - Via I Maggio, 6 - 52040 Terontola (AR) - Tel. 67208

Vendo oltre 350 programmi eccezionali su cassetta per Commodore 64. Prezzi modicissimi, richiedere la lista o inviare la propria per lo scambio. Roberto Borrelli - Via Panoramica, 31 - 60100 Ancona - Tel. 071/56512

Cambio cartuccia "Defender" con qualsiasi Adventure. Annuncio sempre valido. Luigi Russo - Via Flumendosa, 33 - 48100 Ravenna

Vendo VIC 20 più 1 manuale di grafica tridimensionale, 3 libri giochi, 6 cassette di "Program", 1 manuale di "BASIC", 1 cassetta di "Playgames", 3 cassette di veri programmi: il tutto, praticamente nuovo, a sole L. 100.000 (valore reale L. 280.000). Andrea Andrisani - Via Gregorio XIII, 14 - 000167 Roma - Tel. 06/6212578

Vendo per C64, su cassetta, oltre 50 stupendi giochi fra cui Mondial Soccer, Zaxxon, Pole Position, Fort Apocalipse, Defender, Decathlon, ecc. Tutti in Turbo Tape; L. 10.000 in blocco. Per informazioni o lista telefonare o scrivere a: Sergio Di Filippo - Via Riccitelli, 3 - 64100 Teramo - Tel. 0861/51977

Cambio, vendo, compro programmi per C64. Inviate le vostre liste. Rispondo a tutti! Telefonare dalle 18.00 alle 20.30. Gianni Cottoni - Via Strambino, 23 - 10010 Carrone (TO) - Tel. 011/712311

Per Commodore 64 vendo oltre 500 programmi: giochi, utilità varie, gestionali, novità dall'Inghilterra a L. 1.500 cadauno. Matteo Saggese - Via Ugo Inchiostri, 89 - 00143 Roma - Tel. 06/5030534

Vendo VIC 20 più espansione da 3-8-16 Kbyte RAM, 8 giochi su nastro, 2 cartucce al prezzo di L. 500.000 trattabili. Telefonare ore pasti. Salvatore Jaconis - Via Carloni, 5 - 22100 Como - Tel. 031/240148

Vendo VIC 20 con periferiche e software. Massimiliano Favilli - Via Livornese Est, 9 - 56030 Perignano (PI) - Tel. 610069

Vendo e cambio programmi eccezionali a prezzi eccezionali per C64. Telefonare per credere! Possibilmente tratto con residenti in Milano. Alberto La Villa - Via A. Inganni, 74 - 20147 Milano - Tel. 02/4150187

Vendo cassetta con Simons' BASIC, 114 comandi in più sui quali lavorare e giocare sul tuo CBM 64 a sole L. 50.000. Luca Bellettato - Via dei Lillà, 53 - 20089 Rozzano (MI) - Tel. 02/8251907

Vendo, cambio cartuccia "Adventureland", ottime condizioni! Riccardo Mecacci - Via Norma Pratelli, 26 - 58022 Follonica (GR) - Tel. 42747

SUPERCOMMODORE — LIBRI

CEDOLA DI ORDINAZIONE - LIBRI
da compilare e spedire in busta chiusa a
J.*soft* – Viale Restelli, 5 - 20124 Milano - Tel. 6888228

Tecniche di programmazione sul Commodore 64	cod. AJAK038	L. 14.800
Commodore 16: essenziali e routine	cod. AJAC057	L. 26.100

Ordino i seguenti libri per un importo totale di L. .... + L. 2.000
come contributo fisso per spese di spedizione

Cod. .... Cod. .... Cod. ....

Cod. .... Cod. .... Cod. ....

☐ Contanti allegati ☐ Assegno allegato n° ....

....

☐ Ho spedito l'importo a mezzo vaglia postale

☐ Ho versato l'importo sul cc. n° 19445204 intestato a J.*soft* - Milano

☐ Pagherò in contrassegno al postino al ricevimento dei volumi (valido solo per i soci in Italia)

Nome ....

Cognome ....

Via ....

Città .... C.A.P. .... Prov. ....

Se richiesta fattura:
Cod. F. e P. Iva ....

Data ....

Firma ....

Per i residenti all'estero: pagamento anticipato (vaglia o versamento su ns. ccp)

SUPERCOMMODORE è bello, però... (ovvero suggerimenti, idee, critiche, richieste e tutto ciò che vi passa per la testa).

....

....

....

....

....

....

....

Nome ....

Cognome ....

Via .... C.A.P. ....

Città .... Tel. ....

SUPERCOMMODORE

La rubrica INPUT/OUTPUT è gratuita ed aperta a tutti i lettori. Chi desidera comprare, vendere o cambiare hardware o software può inviare il tagliando a J.*soft* - Viale Restelli, 5 - 20124 MILANO

INPUT/OUTPUT

☐ COMPRO ☐ VENDO ☐ CAMBIO

☐ VIC 20 ☐ C 16 ☐ C64 ☐ PERIF. ☐ SOFTWARE

....

....

....

....

....

....

Nome .... Cognome ....

Via .... C.A.P. ....

Città .... Tel. ....

Ritorna in edicola

VIDEO BASIC

Il corso più entusiasmante su cassetta
del Gruppo Editoriale Jackson per **Commodore 64, VIC 20 e Spectrum**

200.000 copie vendute

del 1° fascicolo della prima edizione

Ogni lezione uno spettacolo

Con la 1ª lezione una cassetta giochi compresa nel prezzo





Il corso è composto da:
20 fascicoli + (Quattordicinali)
20 cassette +
5 splendidi raccoglitori

Oggi è davvero facile imparare il Basic. Con Video Basic il corso su cassetta che ti permette di programmare subito il tuo computer. È facile: tu chiedi, lui risponde, tu impari. Passo dopo passo. Sul tuo schermo appaiono le domande, le risposte, gli esercizi e tu, senza fatica, presto e bene, impari a conoscere e programmare il tuo computer, sia esso un VIC 20, un Commodore 64 o un Sinclair.
Video Basic è in edicola.
Provalo subito.
Ogni lezione è uno spettacolo.

Oggi il Basic si impara così. Video Basic, il corso su cassetta per parlare subito col tuo computer.

Video Basic
per imparare non solo il Basic.



Un'altra grande idea firmata
GRUPPO EDITORIALE JACKSON
Milano•San Francisco•Londra•Madrid